Information Management software

Getting Started with IBM DB2 Express-C 9.7 and Parallels Virtuozzo Containers 4.0

26 February 2010

Andre Albuquerque *(andreda@ca.ibm.com), IBM Canada Lab*
Irina Delidjakova *(idelidja@ca.ibm.com), IBM Canada Lab*

CONTENTS

1. Introduction

IBM® DB2® Express-C software ("DB2 Express-C") is a no-charge community edition of the DB2 data server; it is free to use for development and free to deploy and distribute [1]. DB2 Express-C provides an excellent foundation for application development for small and medium businesses. Its quick setup, ease of use, and powerful self-managing capabilities help increase productivity while helping to reduce administration time and costs.

Embodying all of the core features of the more scalable DB2 editions, the DB2 Express-C data server uses up to two processor cores and 2 GB of RAM. Additionally, you can use DB2 Express-C with any storage solution, and there are no restrictions on database size or any other artificial limitations.

Combined with Parallels Virtuozzo Containers 4.0 for Linux®, DB2 Express-C software can be a powerful addition to your information management infrastructure. Virtuozzo Containers offers a high-performance, space-efficient operating system virtualization solution that you can use to create new virtual machines, also known as *containers,* within seconds.

To simplify DB2 data server deployment for users of Virtuozzo Containers for Linux, IBM has made available EZ templates for DB2 Express-C 9.7, optimized specifically for Virtuozzo. An EZ template is a set of application files and registry settings that you install on the host operating system. These files and registry settings are usable by any Virtuozzo environment when you mount the Virtuozzo File System (VZFS). Harnessing the power of the VZFS, you can have a DB2 Express-C data server fully installed, configured, allowing you to provision new test and production servers as needed.

Virtuozzo OS virtualization targets the best management, performance, and efficiency. As Figure 1 shows, the standard host operating system resides in the base, followed by the Virtuozzo virtualization layer for the Windows® or Linux operating systems, which allows the creation of isolated containers on top of the hosting operating system. Each container has a kernel service abstraction layer that ensures the isolation and security of resources between different containers. Finally, the container itself houses the application or workload.

[1] For more information, visit the DB2 Express-C Partner Zone at http://www-01.ibm.com/software/data/db2/express/partners.html.

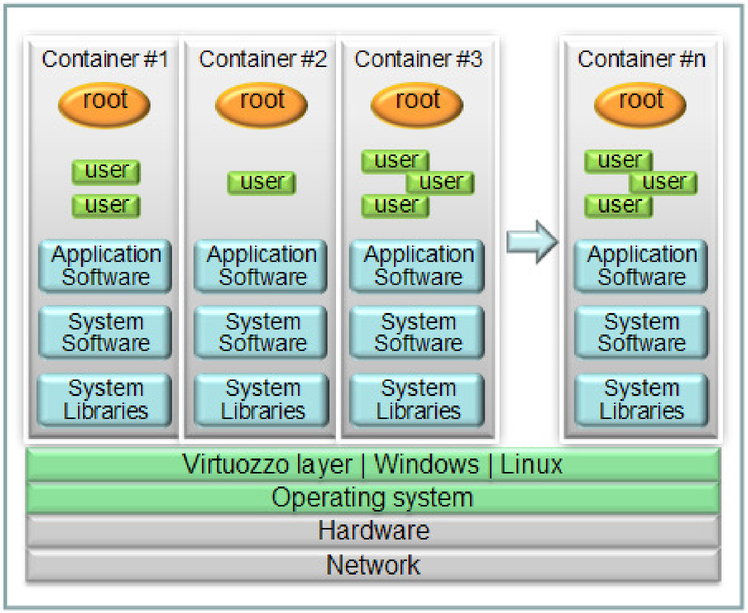


Figure 1. Virtuozzo file system and container overview

This paper will help you get started with DB2 Express-C and Virtuozzo Containers for Linux. You can use this paper to perform the following tasks:

- Install the DB2 Express-C EZ templates on a Virtuozzo hardware node
- Create and configure repositories for DB2 packages for RPM and Debian
- Install and test DB2 Express-C inside a Virtuozzo container
- Troubleshoot and tune a container

2. Prerequisites

Before proceeding, ensure that you have provisioned a Virtuozzo Containers 4.0 hardware node and that it is ready for use[2]. Additionally, read this section carefully to ensure that you adequately prepare your system for DB2 installation.

2.1 Operating system templates

Before creating a container into which you will install DB2 Express-C 9.7, you must cache one or more of the DB2 Express-C 9.7 EZ templates into your repository. EZ templates are available for both 32- and 64-bit versions of the following Linux distributions:

- Red Hat Enterprise Linux 5
- SUSE Linux Enterprise Server 11

[2] For installation and configuration guides, refer to the "Technical Resources" section on the Parallels Virtuozzo Containers page at http://www.parallels.com/products/pvc45/resources/docs/.

- Ubuntu 8.04

These distributions have been validated for compatibility with DB2 data server software. For more information about validated and recommended Linux distributions for DB2 software, see http://www.ibm.com/db2/linux/validate.

2.2 Package dependencies

The DB2 Express-C 9.7 EZ templates automatically resolve the required package dependencies if the aforementioned packages are in your operating system repository or are installed in your container.

3. Installing DB2 EZ templates on a hardware node

Before you can install DB2 Express-C 9.7 in a Virtuozzo container, you must install one or more DB2 EZ templates on the hardware node. These templates are packaged as RPM files and are available for download at http://www.ibm.com/developerworks/wikis/display/im/Parallels%20Virtuozzo%20Support. The files are as follows.

Operating system	File name
RHEL5 x86	DB2-EXPC-redhat-el5-x86-ez-9.7.0-1.noarch.rpm
RHEL5 x86_64	DB2-EXPC-redhat-el5-x86_64-ez-9.7.0-1.noarch.rpm
SLES 11 x86	DB2-EXPC-sles-11-x86-ez-9.7.0-1.noarch.rpm
SLES 11 x86_64	DB2-EXPC-sles-11-x86_64-ez-9.7.0-1.noarch.rpm
Ubuntu 8.04 x86	DB2-EXPC-ubuntu-8.04-x86-ez-9.7.0-1.noarch.rpm
Ubuntu 8.04 x86_64	DB2-EXPC-ubuntu-8.04-x86_64-ez-9.7.0-1.noarch.rpm

After downloading a template, install it by issuing the following command on the hardware node:

```
# rpm -ivh template_file
```

You can verify the installation of your DB2 EZ templates by issuing the following command:

```
# rpm -qa | grep db2
```

This command lists all DB2 EZ templates that you have installed on the hardware node.

4. Creating and Configuring a DB2 repository

After you install the DB2 EZ templates on the hardware node, you must configure Virtuozzo to pull DB2 packages from a repository. For RHEL- and SUSE-based containers, you must configure a local repository. If you are installing DB2 on an Ubuntu operating system, you have two options:

- Configure the hardware node (host server) to cache the DB2 9.7 Debian binary files directly from the Ubuntu software repository[3]
- Set up a local repository for the DB2 binary files for Debian on the hardware node

Each of these tasks is detailed in the following sections.

4.1 Configuring the hardware node to cache the DB2 binary files for Debian from the Ubuntu software repository

To configure the host server to automatically cache the DB2 EZ templates for the Ubuntu operating system, you must add the address of the Canonical partner repository to the `/etc/vztt/url.map` file. This file contains a list of repositories that are used by the Virtuozzo containers. To point to the repository, add the following line to the file:

```
$DB2DEBSERVER http://archive.canonical.com
```

4.2 Using the DB2 binary files for Debian and RPM from a local repository

If you want to use your own local repository, you must download all necessary DB2 packages, set up the local repository, and then configure Virtuozzo to use this repository. This section details all steps that are required to use a DB2 repository.

4.2.1 Obtaining DB2 packages

[3] For more information about the Ubuntu software repository, see http://www.ubuntu.com/community/ubuntustory/components.

The first step in creating a DB2 repository is to obtain the software binary packages for each platform on which you will install DB2. The packages are available at **http://www.ibm.com/developerworks/wikis/display/im/Parallels%20Virtuozzo%20Support**. The following table lists the names of the available packages.

DB2 Binary Files	
Platform	**File name**
32-bit versions of RHEL, SUSE	`IBM-DB2-EXPC-latest.i586.rpm`
64-bit versions of RHEL, SUSE	`IBM-DB2-EXPC-latest.x86_64.rpm`
32-bit versions of Ubuntu	`db2exc_9.7.1-1_i386.deb`
64-bit versions of Ubuntu	`db2exc_9.7.1-1_amd64.deb`

DB2 Init Script Files	
Operating System	**File name**
RHEL	`db2run-9.7-RHEL-latest.noarch.rpm`
SUSE	`db2run-9.7-SLES-latest.noarch.rpm`

As the name suggests, init scripts are initialization scripts used by the operating system to start the DB2 program, and activate all databases and set all necessary kernel parameters after a system power up and to stop the DB2 program before a system shutdown / reboot. On Debian-based containers, one RPM file contains both the base DB2 installation and DB2 init scripts, while on RPM-based containers, the RPM init script file is separate from the DB2 installation RPM file, as noted in the previous table. To find out more about the DB2 init scripts, see step 2; "Verify that the DB2 database server is running." (in section 5.3).

4.2.2 Configuring a DB2 repository for RPM-based Linux distributions

This section describes how to configure a DB2 repository for RPM packages in a container by using the `fedora-core-9-x86_64` EZ template. The commands shown in the following steps are examples. If you choose to use another operating system in your repository container, the process is similar, but certain paths might differ. Also, you might want to specify other values that are different from those shown.

1. On the hardware node, create the container that will host the repository by issuing the **vzctl create** command:

```
# vzctl create 500 --ostemplate fedora-core-9-x86_64 --config slm.2048MB --
ipadd 172.16.50.50 --hostname DB2RPMREPO
```

   This command creates a container with Virtuozzo Environment ID (VEID) 500 and assigns IP address 172.16.50.50 and host name DB2RPMREPO to the container. If VEID 500 is already in use in your system, you must select a different VEID. Also, if the IP address 172.16.50.50 is already in use by another container or if you are using a subnet other than 172.16.0.0, you must modify the IP address accordingly when you specify the command.

2. Set the root password for the container by issuing the **vzctl set** command:

```
# vzctl set 500 --userpasswd root:password
```

   In this case, the root password for the container is set to password.

3. Assign the container adequate disk space to host the DB2 packages by issuing the **vzctl set** command:

```
# vzctl set 500 --diskspace 1G --save
```

   In this case, the container is given a soft disk space limit of 1 GB.

4. Start the container by issuing the **vzctl start** command:

```
# vzctl start 500
```

5. On the hardware node, create the five directories in the container that will contain the DB2 packages by issuing the following commands:

```
# mkdir -p /vz/root/500/var/www/html/download/ibm/db2/x86/base
# mkdir -p /vz/root/500/var/www/html/download/ibm/db2/x86/updates
# mkdir -p /vz/root/500/var/www/html/download/ibm/db2/x86_64/base
# mkdir -p /vz/root/500/var/www/html/download/ibm/db2/x86_64/updates
# mkdir -p /vz/root/500/var/www/html/download/ibm/db2/noarch
```

   Note that /var/www/html/ is the default path to the public Web root in Fedora Core 9. If you are using a different operating system in your repository container, you must change the paths for the commands accordingly.

6. Copy the DB2 packages that you obtained in the section 3 “Installing DB2 EZ templates on a hardware to the appropriate locations by issuing the following commands:

```
# cp IBM-DB2-EXPC-latest.i586.rpm \
/vz/root/500/var/www/html/download/ibm/db2/x86/base
# cp IBM-DB2-EXPC-latest.x86_64.rpm \
/vz/root/500/var/www/html/download/ibm/db2/x86_64/base
# cp db2run-9.7-SLES-latest.noarch.rpm \
/vz/root/500/var/www/html/download/ibm/db2/noarch
# cp db2run-9.7-RHEL-latest.noarch.rpm \
/vz/root/500/var/www/html/download/ibm/db2/noarch
```

7. Install the createrepo package in the hardware node by issuing the **install** command:

```
# vzpkg install 500 -p createrepo
```

8. Create the repository metadata by issuing the following commands:

```
vzctl exec 500 createrepo /var/www/html/download/ibm/db2/x86/base
vzctl exec 500 createrepo /var/www/html/download/ibm/db2/x86/updates
vzctl exec 500 createrepo /var/www/html/download/ibm/db2/x86_64/base
vzctl exec 500 createrepo /var/www/html/download/ibm/db2/x86_64/updates
vzctl exec 500 createrepo /var/www/html/download/ibm/db2/noarch
```

9. In the /etc/vztt/url.map file, add the variable **$DB2SERVER**, and set it to the IP address of the repository container. (See the step 1 "On the hardware node, create the container that will host the repository".)

   For example, if your repository container uses the IP address 172.16.50.50, add the following line to the /etc/vztt/url.map file:

   $DB2SERVER http://172.16.50.50

10. On the hardware node, issue the following command to start the Web server in the container and set the Web server to start automatically at boot time:

```
# vzctl exec 500 service httpd start
```

   Depending on the operating system that you use in the repository container, the commands to start the http server and set it to start automatically at boot time might differ from those shown.

   You now have a correctly configured local repository serving DB2 packages for RPM for both 32- and 64-bit versions of the RHEL and SUSE Linux distributions.

4.2.3 Configuring a DB2 repository for Debian-based operating systems

This section describes how to configure a DB2 repository for Debian-based environments in a container by using the ubuntu-8.04-x86_64 EZ template. The commands shown in the following steps are examples. If you choose to use another operating system to configure the repository, the process is similar, but certain paths might differ. Also, you might want to specify other values that are different from those shown.

1. On the hardware node, to create the container that will host the repository, issue the following command:

```
# vzctl create 501 --ostemplate ubuntu-8.04-x86_64 --config slm.2048MB --
ipadd 172.16.50.51 --hostname DB2DEBREPO
```

   This command creates a container with VEID 501 and assign IP address 172.16.50.51 to the container.

2. Set the root password for the container by issuing the **vzctl set** command:

```
# vzctl set 501 --userpasswd root:password
```

   In this case, the root password for the container is set to `password`.

3. Assign the container adequate disk space (usually, each DB2 binary file has 400 MB) to host the DB2 packages by issuing the **vzctl set** command:

```
# vzctl set 501 --diskspace 1G --save
```

   In this case, the container is given a soft disk space limit of 1 GB.

4. Start the container by issuing the **vzctl start** command:

```
# vzctl start 501
```

5. On the hardware node, create the eight directories in the container that will contain the DB2 packages by issuing the following commands:

```
# mkdir -p /vz/root/501/var/www/ubuntu/binary-i386
# mkdir -p /vz/root/501/var/www/ubuntu/updates-i386
# mkdir -p /vz/root/501/var/www/ubuntu/binary-amd64
# mkdir -p /vz/root/501/var/www/ubuntu/updates-amd64
# mkdir -p /vz/root/501/var/www/ubuntu/dists/hardy/partner/binary-i386
# mkdir -p /vz/root/501/var/www/ubuntu/dists/hardy/partner/updates-i386
# mkdir -p /vz/root/501/var/www/ubuntu/dists/hardy/partner/binary-amd64
# mkdir -p /vz/root/501/var/www/ubuntu/dists/hardy/partner/updates-amd64
```

   Note that `/var/www/` is the default path to the public Web root in Ubuntu 8.04. If you are using a different operating system in your repository container, you must change the paths in the commands accordingly.

   The directory structure that is shown for the previous commands was reproduced from the Canonical partner repository, which contains the most current DB2 packages for Debian. The structure allows you to choose between downloading the Debian packages into your container from a locally created repository or directly from the Canonical partner repository. Instructions on how to update the `url.map` file are given in section 4.1" of this document.

6. Copy the DB2 packages to the `binaries` directory by issuing the following commands:

```
# cp db2exc_9.7.1-1_i386.deb /vz/root/501/var/www/ubuntu/binary-i386
# cp db2exc_9.7.1-1_amd64.deb /vz/root/501/var/www/ubuntu/binary-amd64
```

7. Install the `dpkg-dev` package into the repository container by issuing the **vzctl install** command:

```
# vzpkg install 501 -p dpkg-dev
```

8. Enter into the container by issuing the **vzctl enter** command:

```
# vzctl enter 501
```

9. Navigate to the DB2 binaries directory, and create the local repository by issuing the **dpkg-scanpackages** command:

```
# cd /var/www/ubuntu
# dpkg-scanpackages binary-i386 /dev/null > dists/hardy/partner/binary-
i386/Packages
# dpkg-scanpackages updates-i386 /dev/null > dists/hardy/partner/updates-
i386/Packages
# dpkg-scanpackages binary-amd64 /dev/null > dists/hardy/partner/binary-
amd64/Packages
# dpkg-scanpackages updates-amd64 /dev/null > dists/hardy/partner/updates-
amd64/Packages
```

10. Compress the `Packages` files by issuing the **gzip** command:

```
# gzip dists/hardy/partner/binary-i386/Packages
# gzip dists/hardy/partner/updates-i386/Packages
# gzip dists/hardy/partner/binary-amd64/Packages
# gzip dists/hardy/partner/updates-amd64/Packages
```

11. Start the Apache Web server in the container and set it to start automatically at boot time by issuing the following commands within the shell of the container:

```
# /usr/sbin/apache2ctl start
# echo "NO_START=0" > /etc/default/apache2
```

12. Exit the container and return to the hardware node by issuing the **vzctl exit** command:

```
# exit
```

13. In the `/etc/vztt/url.map file`, add the variable **$DB2DEBSERVER,** and set it to the IP address of the repository container. (See ***Error! Reference source not found.***section 4.1”.)

    For example, if your repository container uses the IP address 172.16.50.51, add the following line to the `/vz/template/conf/vztt/url.map file`:

    `$DB2DEBSERVER http://172.16.50.51`

Depending on the operating system that you use in the repository container, the commands to start Apache and set it to start automatically at boot time might differ from those shown.

You now have a correctly configured local repository serving DB2 packages for Debian for both 32- and 64-bit versions of the Ubuntu operating system.

5. Installing, configuring, and testing DB2 software in a container

This section details the process of installing and configuring DB2 software in a container. It also explains how to test the DB2 installation to verify that it is working properly.

5.1 Installing DB2 software in a container

The EZ templates created for the DB2 Express-C data server simplify the installation of DB2 software into your containers. To install DB2 software into a 32-bit or 64-bit container, on the hardware node, issue the following command:

```
# vzpkg install VEID DB2-EXPC
```

`where VEID` is the ID of the container into which you want to install the DB2 software. The platform is defined within the EZ template definition.

The **vzpkg install** command downloads the appropriate DB2 packages from the repositories and installs the DB2 data server into the specified container.

5.2 DB2 software configuration in a container

The EZ templates perform all configurations after you install the DB2 software.

The following system groups are created during the post-installation steps:

- Instance user group db2iadm1
- Fenced user group db2fadm1
- Database Administration Server (DAS) user group dasadm1

The following system users are created during the post-installation steps:

- Instance user db2inst1
- Fenced user db2fenc1
- DAS user dasusr1

The DB2 Express-C license is then registered into the installation, and the required DB2 symlinks are created.

As a final step, the DB2 init script is invoked to start the DAS and the created instance user.

5.3 Testing the DB2 configuration and creating a SAMPLE database

After you install the DB2 Express-C data server and it is configured, it is a good idea to run a quick test to ensure that everything works as it should.

1. Connect to your container as root by using SSH. For example, to connect to a container listening on IP address 172.16.10.10, issue the following command:

```
# ssh root@172.16.10.10
```

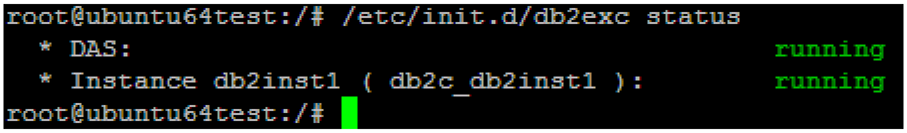
2. Verify that the DB2 database server is running.

   Virtuozzo containers that have DB2 installed are equipped with a db2 service, which automatically starts DB2 during the boot sequence and stops before resetting or shutting down the server. To see the status of the DB2 database manager, issue one of the following commands:

   - On an Ubuntu-based container, issue the following command:

```
# /etc/init.d/db2exc status
```

   The output should be similar to this:

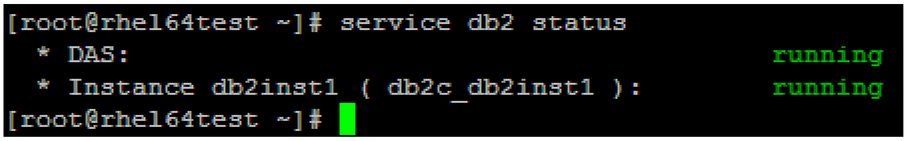


```
root@ubuntu64test:/# /etc/init.d/db2exc status
  * DAS:                                          running
  * Instance db2inst1 ( db2c_db2inst1 ):          running
root@ubuntu64test:/#
```

   - On an RHEL-based container, issue the following command:

```
# service db2 status
```

   The output should be similar to this:

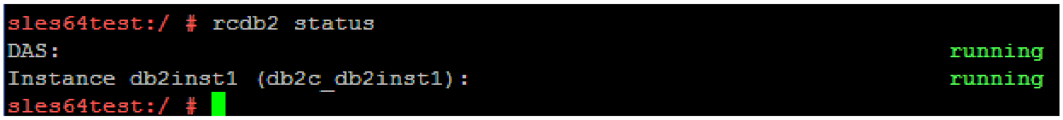


```
[root@rhel64test ~]# service db2 status
  * DAS:                                          running
  * Instance db2inst1 ( db2c_db2inst1 ):          running
[root@rhel64test ~]#
```

   - On a SUSE-based container, execute the following command:

```
# rcdb2 status
```

   The output is similar to this:

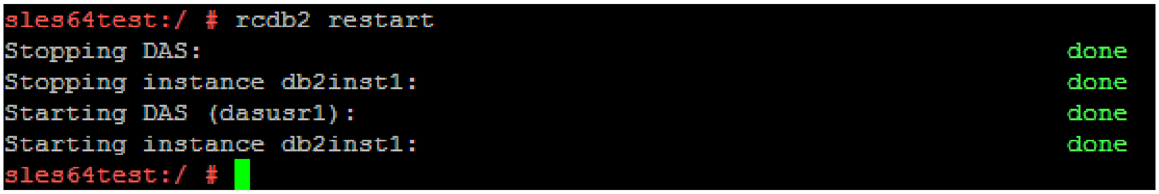


```
sles64test:/ # rcdb2 status
DAS:                                                                   running
Instance db2inst1 (db2c_db2inst1):                                     running
sles64test:/ #
```

Other options of the DB2 init script include **start**, **stop**, and **restart**. For example, on a SUSE container, you can restart the DB2 database manager by running the following command:

```
# rcdb2 restart
```

The output is similar to what follows:



```
sles64test:/ # rcdb2 restart
Stopping DAS:                                                        done
Stopping instance db2inst1:                                          done
Starting DAS (dasusr1):                                              done
Starting instance db2inst1:                                          done
sles64test:/ #
```

3. Change to the instance user, as shown in the following example. If this is the first time that you installed the DB2 EZ template into the container, the instance user is db2inst1.

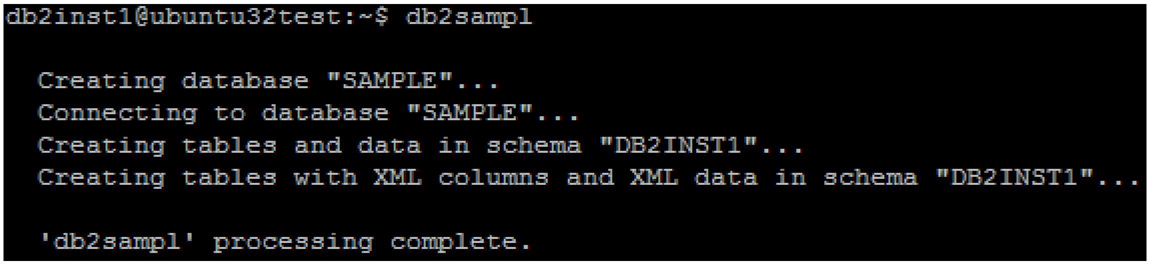
```
# su – db2inst1
```

   Each time that you reinstall DB2 software into the same container, the number in the instance user name is incremented. (The number is incremented regardless of whether you replace the software or update it with a fix pack, which is packaged as an RPM binary file). For example if db2inst1 exists, the new instance user name is db2inst2. This mechanism preserves any data that existed within previous DB2 installations.

4. Test the integrity of the installation by creating a sample database. To create the database, issue the following command:

```
# db2sampl
```

A database named SAMPLE that contains SQL and XML data is created. The successful creation of this database shows that the basic functionality of the DB2 data server is present, as you can see in the following example:

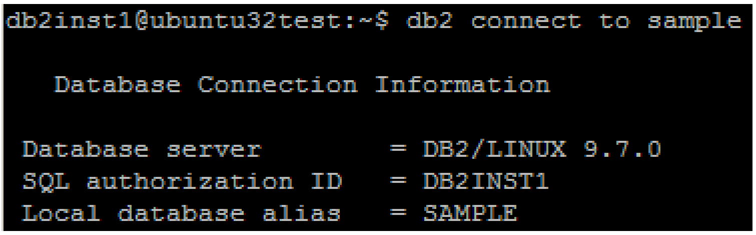


```
db2inst1@ubuntu32test:~$ db2sampl

  Creating database "SAMPLE"...
  Connecting to database "SAMPLE"...
  Creating tables and data in schema "DB2INST1"...
  Creating tables with XML columns and XML data in schema "DB2INST1"...

  'db2sampl' processing complete.
```

5. Create a connection to the sample database by issuing the following command:

```
# db2 connect to sample
```

The following connection information is shown:

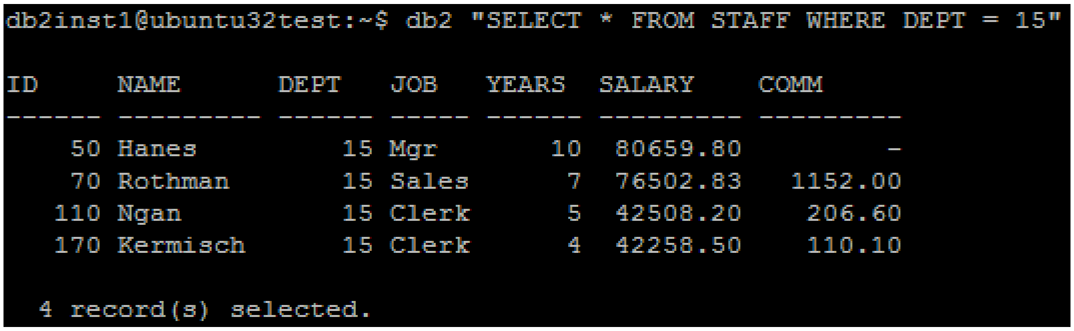


```
db2inst1@ubuntu32test:~$ db2 connect to sample

   Database Connection Information

 Database server        = DB2/LINUX 9.7.0
 SQL authorization ID   = DB2INST1
 Local database alias   = SAMPLE
```

6. To test the integrity of the created database, execute some SQL commands. For example, you can query one of the tables and inspect the results:

```
# db2 “SELECT * FROM STAFF WHERE DEPT = 15”
```

The following screen capture shows the expected returned data:



```
db2inst1@ubuntu32test:~$ db2 "SELECT * FROM STAFF WHERE DEPT = 15"

ID     NAME      DEPT   JOB   YEARS  SALARY    COMM
------ --------- ------ ----- ------ --------- ---------
    50 Hanes         15 Mgr       10  80659.80         -
    70 Rothman       15 Sales      7  76502.83   1152.00
   110 Ngan          15 Clerk      5  42508.20    206.60
   170 Kermisch      15 Clerk      4  42258.50    110.10

  4 record(s) selected.
```

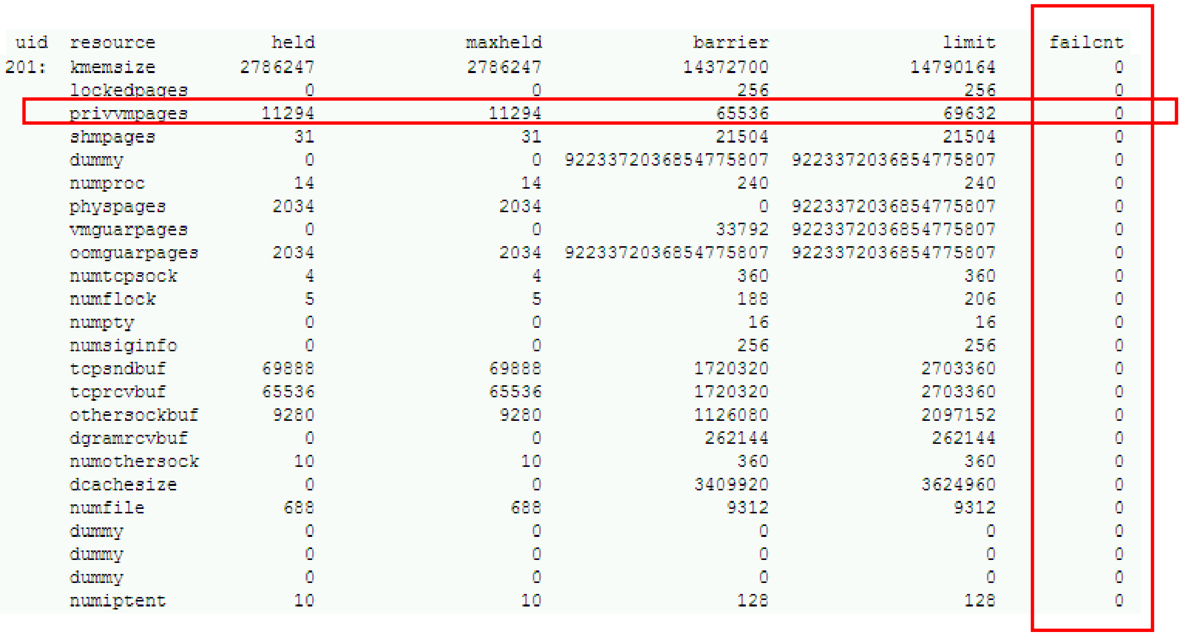
You have now completed the DB2 software setup for Virtuozzo Containers 4.0.

6. Tuning and troubleshooting

If you encounter any errors while using DB2 in a Virtuozzo container or if you find performance to be lagging, it is a good idea to examine the assigned resources for your container and determine how much of these resources have been used.

The `/proc/user_beancounters` file shows the available resources per container. You can use this file to ensure that no resource allocation failures occurred for any of the containers.

Consider the following example:



```
uid   resource            held              maxheld              barrier                limit    failcnt
201:  kmemsize         2786247              2786247             14372700             14790164          0
      lockedpages            0                    0                  256                  256          0
      privvmpages        11294                11294                65536                69632          0
      shmpages              31                   31                21504                21504          0
      dummy                  0                    0  9223372036854775807  9223372036854775807          0
      numproc               14                   14                  240                  240          0
      physpages           2034                 2034                    0  9223372036854775807          0
      vmguarpages            0                    0                33792  9223372036854775807          0
      oomguarpages        2034                 2034  9223372036854775807  9223372036854775807          0
      numtcpsock             4                    4                  360                  360          0
      numflock               5                    5                  188                  206          0
      numpty                 0                    0                   16                   16          0
      numsiginfo             0                    0                  256                  256          0
      tcpsndbuf          69888                69888              1720320              2703360          0
      tcprcvbuf          65536                65536              1720320              2703360          0
      othersockbuf        9280                 9280              1126080              2097152          0
      dgramrcvbuf            0                    0               262144               262144          0
      numothersock          10                   10                  360                  360          0
      dcachesize             0                    0              3409920              3624960          0
      numfile              688                  688                 9312                 9312          0
      dummy                  0                    0                    0                    0          0
      dummy                  0                    0                    0                    0          0
      dummy                  0                    0                    0                    0          0
      numiptent             10                   10                  128                  128          0
```

To determine whether resource allocation failed, you can look at the `failcnt` column. This column shows the number of failed requests for increased resources. For example, in the sample output, the limit for the `privvmpages` parameter is 69,632. If the container requests more than this limit value, the resource request is denied, and the value of the `failcnt` column is incremented.

A non-zero value in the `failcnt` column of a parameter signifies that resource requests are failing and that you need to immediately increase the barrier and limit values. Otherwise, container performance and stability might be affected. However, it is also important to note that the value of the `failcnt` column does not reset to zero until you restart a container. Thus, if you increase the barrier and limit values for a parameter that is associated with failed requests, do not be surprised when its `failcnt` value remains the same. The important thing is that the value of `failcnt` does not increase.

7. Getting assistance

We hope that this paper has been useful in helping you to get started with DB2 Express-C 9.7 and Parallels Virtuozzo Containers 4.0 for Linux.

For more information about IBM DB2 data server software, refer to the following resources:

- The IBM DB2 9.7 for Linux, UNIX, and Windows Information Center: **http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp.** This site contains documentation related to DB2 9.7 for Linux, UNIX®, and Windows, including information about DB2 commands and data objects.

- The IBM developerWorks® forum for IBM DB2 Express: **http://www.ibm.com/developerworks/forums/forum.jspa?forumID=805**. This forum hosts a community of helpful DB2 users and experts, many of whom are willing to help answer any questions that you might have.

For more information about Parallels Virtuozzo Containers, refer to the following resources:

- Virtuozzo documentation in PDF and HTML formats:
  http://www.parallels.com/en/products/virtuozzo/docs/.

- The following high-use forum for Virtuozzo Containers, which you might find helpful if you are experiencing problems with Virtuozzo configuration or usage:
  http://forum.parallels.com/forumdisplay.php?f=160

Finally, you can also direct questions pertaining to this documentation to **askdata@ca.ibm.com.**

IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada

Printed in Canada
03/2010

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow a disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The information in this publication is provided AS IS without warranty. Such information was obtained from publicly available sources, is current as of March, 2010, and is subject to change. Any performance data included in the paper was obtained in the specific operating environment and is provided as an illustration. Performance in other operating environments may vary. More specific information about the capabilities of products described should be obtained from the suppliers of those products.