

Using VMware vFabric Postgres

vFabric Postgres 9.3.5

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see <http://www.vmware.com/support/pubs>.

EN-001393-00

vmware[®]

You can find the most up-to-date technical documentation on the VMware Web site at:

<http://www.vmware.com/support/>

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Copyright © 2012–2014 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

Preface	5
1 VMware Customizations for PostgreSQL	7
vFabric Postgres Virtual Appliance Enhancements	7
Deploying vFabric Postgres	8
Passwords in vFabric Postgres	8
2 Installing vFabric Postgres	11
Installation Overview	11
System Requirements	12
Deploy the OVA File	13
Install vFabric Postgres Using RPM Files	14
Install vFabric Postgres as a Windows Service	15
Uninstall the vFabric Postgres Windows Service	17
Install vFabric Postgres on vCloud Hybrid Service	18
3 vFabric Postgres Client Tools and Libraries	19
Overview of Tools and Libraries	19
Client Tool Packages and Drivers	20
Install the Client Tools Package	21
Add an x86 vFabric Postgres ODBC Data Source on Windows	22
Relink Your Application with vFabric Postgres libpq	23
4 Managing vFabric Postgres	25
Migrate PostgreSQL Data from Earlier Versions Into vFabric Postgres 9.3	25
Migrate PostgreSQL Data Into vFabric Postgres	26
Restarting the vFabric Postgres Service	26
Connection to a vFabric Postgres Database	27
Accounts and Services	27
Safeguarding Data	28
About vFabric Postgres Replication	31
Create a Replication User Account	32
Create a Replica Server	33
Promote a Replica Database to Primary Database	34
Monitoring Replication Status	34
Using Perl and Python Language Extensions	35
Viewing Performance Statistics	36
Troubleshooting Guidelines	38
5 Using the Graphical User Interface	39
Deploy the Graphical User Interface	39

Access the Graphical User Interface 39

Database Entity Management 40

SQL Management 45

Index 47

Preface

Using VMware vFabric Postgres provides information about installing and using a VMware vFabric Postgres Standard Edition DBMS.

Intended Audience

This information is intended for anyone who wants to install or use a vFabric Postgres Standard Edition DBMS. The information is written for experienced Windows or Linux system administrators who are familiar with virtual machine technology and datacenter operations.

Related Publications

The vFabric Suite documentation has information about the components of the vFabric suite.

For information about managing vFabric Postgres databases, see the public PostgreSQL documentation at <http://www.postgresql.org/docs/>. Because vFabric Postgres is compatible with PostgreSQL, you can manage vFabric Postgres databases using the information in that documentation.

To access the current versions of VMware documentation, go to <http://www.vmware.com/support/pubs>.

VMware Customizations for PostgreSQL

1

VMware vFabric Postgres is an ACID-compliant, ANSI-SQL-compliant transactional, relational database management system that is designed for the virtual environment and optimized for vSphere. It is based on the PostgreSQL open-source relational database and is compatible with PostgreSQL.

vFabric Postgres databases are managed by a DBMS that consists of a server and a client. VMware supports all standard PostgreSQL connection tools and interaction methods, and optionally allows you to use a GUI tool for database management.

This chapter includes the following topics:

- [“vFabric Postgres Virtual Appliance Enhancements,”](#) on page 7
- [“Deploying vFabric Postgres,”](#) on page 8
- [“Passwords in vFabric Postgres,”](#) on page 8

vFabric Postgres Virtual Appliance Enhancements

VMware vFabric Postgres virtual appliance includes features that are not available with the open source PostgreSQL DBMS.

The virtual appliance include the following enhancements

Automatic Tuning

If you deploy the vFabric Postgres appliance, associated vFabric Postgres databases have higher default values than standard PostgreSQL databases for many critical settings, including `shared_buffers`, `checkpoint_segments`, and `wal_buffers`. The higher default values improve out-of-box

vFabric Postgres performance with a slight increase in disk space and memory requirements. The result is that users of an embedded vFabric Postgres database can more easily tune the database for their workload.

If you are using vFabric Postgres, and you use the RPM installation, these changes to default values are not made.

Separate XLog Files and Archive Files

Files of `pg_xlog` are stored on a different partition than that used by the main database to better leverage I/O activity on the server, and provide strict control over the storage space allowed for WAL files. This strategy is used as well for the archive to cover cases where a partition disk failure would prevent the reuse of archived WAL files for recovery. WAL archiving is enabled on a given node when a standby node is enabled using the replication scripts that connect to this node.

Replication Scripts

A set of scripts that simplify the management of replication between vFabric Postgres nodes: slave creation, node promotion, and replication monitoring. The scripts are located in the folder `/opt/vmware/vpostgres/current/scripts`.

PGDATA Environment Variable

The virtual appliance provides the PGDATA environment variable, which specifies the directory to use for data storage. The value is set to `/var/vmware/vpostgres/current/pgdata`.

Deploying vFabric Postgres

You can deploy a vFabric Postgres DBMS as a virtual appliance (OVA file) or by using RPM packages.

- You can deploy the virtual appliance to create a virtual machine with the operating system (SLES 11, SP 2 64-bit Linux), a vFabric Postgres server, and a vFabric Postgres client preinstalled. The appliance version of the vFabric Postgres database includes VMware virtualization technology.
- You can use RPM to deploy vFabric Postgres. Use RPM installation with a physical host, or create a virtual machine and install one of the supported operating system that are listed on the datasheet. Use `-ivh` commands to install the RPMs. You can use this method to install the vFabric Postgres server and client software.
- You can use a Microsoft Windows Installer (MSI) to install vFabric Postgres on a Windows physical host, or create a virtual machine using a supported Windows operating system. The software is installed as a Windows service.

Passwords in vFabric Postgres

You specify the passwords for vFabric Postgres users during deployment, and can change passwords for each user individually at a later time.

Passwords differ slightly depending on whether you deploy the OVF or perform an RPM installation.

OVA Deployment

After deployment of a vFabric Postgres OVA, three users are defined.

Table 1-1. vFabric Postgres Users for OVF Deployment

User Name	User Type
root	operating system user
postgres	operating system user
postgres	database user

You can specify a single password for all three users during deployment of the OVA. If you did not specify a password during deployment, you are prompted for a password when you access the virtual appliance console for the first time.

NOTE Remote login for root operating system user is disabled. If you use SSH to log in you must do so as the postgres operating system user.

For automated deployments, you can use `ovftool --prop:Password=secret`.

RPM Installation

During RPM installation, the installer creates the following users.

Table 1-2. vFabric Postgres Users for RPM installation

User Name	User Type
postgres	operating system user
postgres	database user

With RPM installation, no initial password is set for the postgres operating system user or the postgres database user. The root user already exists before the RPM installation and its password is set using Linux commands.

- To set the postgres operating system user password, log in as root and use the `passwd` command to set a new password.

```
passwd postgres
```

- To set the postgres database user password, log in as the postgres operating system user and use the `alter` command.

```
/opt/vmware/vpostgres/current/bin/psql -c "alter user postgres with password 'your-  
password'"
```

Changing Passwords After Installation

After installation, you can change passwords as follows.

- Use the `/opt/aurora/sbin/set_password` command to change the password for all three users.
- Use the `passwd` command to change passwords individually for the system users.
- Use the following command to change the password for the postgres database user.

```
/opt/vmware/vpostgres/current/bin/psql -c "alter user postgres with password 'your-  
password'"
```


Installing vFabric Postgres

Before you install vFabric Postgres, review the requirements and the deployment or installation process.

This chapter includes the following topics:

- [“Installation Overview,”](#) on page 11
- [“System Requirements,”](#) on page 12
- [“Deploy the OVA File,”](#) on page 13
- [“Install vFabric Postgres Using RPM Files,”](#) on page 14
- [“Install vFabric Postgres as a Windows Service,”](#) on page 15
- [“Uninstall the vFabric Postgres Windows Service,”](#) on page 17
- [“Install vFabric Postgres on vCloud Hybrid Service,”](#) on page 18

Installation Overview

The vFabric Postgres server and client software are distributed together. You can either deploy an Open Source Virtual Appliance (OVA) file or install a series of RPM packages.

Virtual Appliance Deployment Overview

The process of deploying a vFabric Postgres virtual appliance is similar on all the different supported virtualization platforms.

- 1 Install one of the VMware virtualization platforms such as vSphere 5.1 or later, VMware Workstation 9.x, VMware Fusion 5.x, or VMware Player 5.x.

NOTE For a production system, only vSphere 5.1 or later is supported.

- 2 Deploy the virtual appliance.
- 3 To manage the new DBMS, log in to the virtual appliance console and use the preinstalled psql tool or point your Web browser to https://your_vApp_IP:8443.
- 4 To manage the virtual appliance, log in to the virtual appliance console or point your Web browser to https://your_vApp_IP:5480.

RPM Deployment Overview

The process of installing the vFabric Postgres DBMS from RPM packages consists of the following high-level tasks.

- 1 Make sure the host or virtual machine that you want to use is running a supported operating systems and meets all the other requirements.
- 2 Download and install the client, server, and init RPM files.
 - a client package
 - b server package
 - c init package
- 3 Log in to the new DBMS using the client software.

System Requirements

You can deploy the virtual appliance and install the RPM packages on several operating systems.

Supported Platforms for OVA Deployment

For the virtual appliance (OVA), several virtualization platforms are supported during development, but support is more limited during production.

Development	While you develop your application and run tests, you can deploy the virtual appliance on the latest edition of any VMware virtualization platform, including VMware vSphere 5.1 or later, VMware Workstation 9.x, VMware Fusion 5.x, or VMware Player 5.x.
Production	In a production environment, you must install vFabric Postgres on VMware vSphere 5.1 ESXi or later.

Resource Requirements for RPM Installation

If you install the RPM files, you must have a physical host or a virtual machine that meets the following minimum requirements.

RAM	512 MB
CPUs or vCPUs	1 or more
Disk Space	12 GB

Resource Requirements for OVA Deployment

The virtual appliance requires 1GB of RAM. See the vSphere documentation for information on the required memory overhead.

Operating Systems

The vFabric Postgres server software is currently supported on the following operating systems.

Red Hat Linux	RHEL 6.2 (64 bit) and 6.4 (64 bit)
SUSE Linux	SLES 11 SP 1 (64 bit), SLES 11 SP2 (64 bit), or SLES 11 SP3 (64 bit)
Oracle Enterprise Linux	OEL 6

Database Clients

The vFabric Postgres product includes JDBC, ODBC, and LIBPQ drivers.

Database clients for Windows, Linux, and Mac OS X, in both 32 bit and 64 bit versions, are included.

Many community PostgreSQL clients, such as Npgsql, and pscopg2 are also supported in both 32-bit and 64-bit configurations. Client drivers for Npgsql are included.

Deploy the OVA File

You can deploy the OVA file on vSphere 5.1 ESXi or later for use during development or for production environments. In addition, can deploy the OVA file on VMware Workstation 9.x, VMware Fusion 5.x, or VMware Player 5.x for use during development.

This topic describes the process when you use the vSphere Web Client with vSphere. If you are using the vSphere Client, or one of the other VMware products, the process is similar but the prompts might differ slightly.

Prerequisites

Download the OVA file from the VMware download site.

Procedure

- 1 Connect to a vCenter Server with the vSphere Web Client.
- 2 Right-click an inventory object that is a valid parent object of a virtual machine, such as a datacenter, folder, cluster, resource pool, or host and select **Deploy OVF Template**.
- 3 If prompted, download the client plug-in.
You have to close all browsers to download the plug-in.
- 4 Respond to the wizard prompts.

Screen	Action
Select Source	Specify the location of the OVA file.
Review Details	Review the OVA information.
Accept EULAS	Review and accept the license agreement.
Select name and folder	Specify the name and location for the virtual appliance.
Select storage	Select the storage for the virtual appliance. You can use the pull-down menu to change the disk format.
Setup networks	Map the networks used in the OVF template to networks in your inventory and select
Customize template	Specify the password that you want to use initially for the three users that the OVA file defines. A minimum of six characters is required.
Ready to Complete	Review the settings and click Finish to start deployment.

When deployment completes, the virtual appliance powers on.

- 5 (Optional) If you did not specify a password during deployment, specify it now. Right-click the virtual appliance and select **Open Console** and enter the initial password for all user accounts.
 - a Click **Edit Settings**.
 - b Click the Options tab and select Properties under vApp Options.
 - c Enter the initial password in the text box on the right.

What to do next

You can now manage your vFabric Postgres environment.

- To manage the new DBMS, log in to the virtual appliance console and use the preinstalled `psql` tool or point your Web browser to `https://your_vApp_IP:8443`.
- To manage the virtual appliance, log in to the virtual appliance console or point your Web browser to `https://your_vApp_IP:5480`.

Install vFabric Postgres Using RPM Files

If you want to install vFabric Postgres on a virtual machine or on a physical host, you can use the RPM installation process.

Different editions of the vFabric Postgres server software are supported on different operating systems. See the datasheet for information.

The vFabric Postgres RPM packages are relocatable, meaning they can be installed into a different directory. The relocatable directory paths are `/opt/vmware/vpostgres` and `/var/vmware/vpostgres`. By default, RPM installs packages in these directories. You can override this during the RPM installation using the command `rpm --prefix directory_name`. For example, you can install the packages in `rpm --prefix /usr/opt`.

Prerequisites

- Create a new virtual machine running a supported operating system, or log in to a virtual machine where one of these operating systems is currently running. You can also install the RPMs on a physical host that runs one of the supported operating systems.
- Verify that you have access to the Internet to download the RPM packages.
- If you install 32-bit binaries on a 64-bit system, install compatibility libraries as well. On RHEL6, use `yum install glibc.i686 nss-softokn-freebl.i686`.

Procedure

- 1 Download at a minimum the following ZIP files from the VMware download site.
 - ZIP file for vFabric Postgres
 - ZIP file for the vFabric Postgres client tools and libraries
 - ZIP file for the vFabric Postgres JDBC driver

Optional components, 32-bit client RPMs, and client tools for Windows, Macintosh, ODBC, and JDBC are also available on the download site.

- 2 Unzip the ZIP files and install each of the RPM files using the `rpm -ivh` command, in the order shown below, or install all files at once with a single command.

```
>rpm -ivh
VMware-VMware-Postgres-osslibs-server-version.x86_64.rpm
VMware-VMware-Postgres-osslibs-version.x86_64.rpm
VMware-VMware-Postgres-libs-version.x86_64.rpm
VMware-Postgres-server-version.x86_64.rpm
```

The RPM installation creates the directory `/opt/vmware/vpostgres/`, and the `postgres` user for the database and operating system.

NOTE The `postgres` database user, and the `postgres` operating system user, are two different users accounts.

- 3 Create a folder labeled `pgdata`, and assign it to the `postgres` user.

```
mkdir /var/vmware/vpostgres/9.3/pgdata
chmod 755 /var/vmware/vpostgres/9.3/pgdata
chown postgres /var/vmware/vpostgres/9.3/pgdata
```

- 4 Change to the `postgres` user.

```
su postgres
```

- 5 Initialize the database using the `init` command.

```
cd /opt/vmware/vpostgres/current/bin
./initdb -D /var/vmware/vpostgres/9.3/pgdata
```

- 6 Start the database using either the `pg_ctl` command or the `postgres` command.

- To start the database using the `postgres` command, use the following command syntax.

```
./postgres -D /var/vmware/vpostgres/9.3/pgdata
```

- To start the database using the `pg_ctl` command, use the following command syntax.

```
./pg_ctl start -D /var/vmware/vpostgres/9.3/pgdata
```

- 7 You can now connect to the database as the `postgres` user using the `psql` command.

```
./psql -U postgres
```

- 8 (Optional) Create a database to test the installation.

This example creates the database `test1`.

```
postgres=#CREATE DATABASE test1;
CREATE DATABASE
```

- 9 If you want to you use the GUI, you can download a ZIP file that contains the `vpgdbem.war` file from the vFabric Postgres download site and move the file to the `webapps` directory of your Tomcat server.

You can then use the `vpgdbem` URI to access the GUI. For example, if your Tomcat server is installed for port 8080, you can access the GUI at `http://ipaddress:8080/vpgdbem`.

What to do next

You can set the passwords for the `postgres` operating system and database user. See [“Passwords in vFabric Postgres,”](#) on page 8.

Install vFabric Postgres as a Windows Service

You can install vFabric Postgres using the Microsoft Windows Installer (MSI) on a Windows physical host, or create a virtual machine using a supported Windows operating system.

The MSI installer lets you install the Windows 64-bit (x64) version of vFabric Postgres as a Windows service. The supported operating systems are Windows 2008, 2008 R2, and Windows 2012.

You can run the MSI installer either by double-clicking the executable file on your Windows desktop, or by running it from the command prompt and supplying custom installation parameters. If you choose to install vFabric Postgres by double-clicking the executable file, the installation uses the default values for directory paths, the database user name and password, and the database name.

A single vFabric Postgres database server (identified by a Windows service name) can serve multiple databases. Each database is identified using a unique database name. By default, when you create a vFabric Postgres database server instance, a system database labeled `postgres` is created. Do not use the `postgres` system database to store your application's data. Instead, create additional databases for use with your applications.

Prerequisites

- Download the MSI installer from the VMware download site.
The installer name is `VMware-Postgres-9.3.5.0-xxxx-x64-CIS.msi`, where `xxxx` is the build number.
- You must run the MSI installer as the Administrator for the Windows operating system.

NOTE You cannot run the installer as a user that is a member of the administrators group. You must run the MSI installer using the default Administrator account.

Procedure

- 1 Log in as the Windows Administrator.
- 2 Open a Windows command prompt on the virtual machine or physical host on which you are going to install vFabric Postgres.
- 3 Change your working directory to the location of the installation executable file.
- 4 From the command line of the virtual machine or physical host where you are installing vFabric Postgres, run the appropriate command string to start the installation.

```
msiexec /i package PG_DATA_DIR="directory" PG_LOG_DIR="directory" PG_PORT=port_number
DB_OWNER=user_name DB_OWNER_PASS="password" DB_NAME=name
```

Option	Description
/i	Installs or configures a product.
package	Specifies the name of the Windows Installer package file. For vFabric Postgres this is: <code>VMware-Postgres-9.3.5.0-xxxx-x64-CIS.msi</code>
PG_DATA_DIR="directory"	(Optional) Specifies the directory to use for the data. If you do not specify a directory, the default is <code>C:\ProgramData\VMware\Postgres\data</code> . Encapsulate the directory you specify in double quotes, which the system discards.
PG_LOG_DIR="directory"	(Optional) Specifies the directory to use for the log file. If you do not specify a directory, the default is <code>C:\ProgramData\VMware\Postgres\log</code> . Encapsulate the directory you specify in double quotes, which the system discards.
PG_PORT=port_number	Specifies the TCP/IP port on which vFabric Postgres listens for connections from client applications. The installation defaults to the value of the <code>PG_PORT</code> environment variable, or, if <code>PG_PORT</code> is not set, it uses the value defined during compilation, which is typically port 5432. If you specify a port other than the default port, all client applications must specify the same port using either command-line options or <code>PG_PORT</code> .
DB_OWNER=db_owner_name	(Optional) Specifies the database owner. Note that the database owner name you specify is not the same as the operating system user account name. The database owner is used for authentication, authorization, and the ownership of schema objects. This user can be mapped to an operating system user when vFabric Postgres is configured to use Windows authentication. The database owner name you specify must be a properly formed SQL identifier. SQL identifiers must begin with a letter (a-z, but also letters with diacritical marks and non-Latin letters) or an underscore (<code>_</code>). Subsequent characters in an identifier or key word can be letters, underscores, and digits (0-9). Dollar signs (<code>\$</code>) are not allowed in identifiers.

Option	Description
DB_OWNER_PASS="password"	<p>(Optional) This optional parameter lets you specify the DB_OWNER user's password. If you do not specify a password, DB_OWNER user's password is generated with a random string.</p> <p>Encapsulate the password you specify in double quotes, which the system discards.</p> <p>The password is stored in %APPDATA%\postgresql\pgpass.conf where %APPDATA% points to the Windows administrator's application data directory (as defined by the APP_DATA environment variable) running the vFabric Postgres MSI installer.</p>
DB_NAME=name	<p>(Optional) The name of the new database to create.</p> <p>The database name you specify must be a properly formed SQL identifier. SQL identifiers must begin with a letter (a-z, but also letters with diacritical marks and non-Latin letters) or an underscore (_). Subsequent characters in an identifier or key word can be letters, underscores, and digits (0-9). Dollar signs (\$) are not allowed in identifiers.</p>

Example: Custom Installation

This example installs vFabric Postgres using the default directory locations for the data and log files. The database owner name is **demo_user**, and the database name is **demo_db**.

The DB_NAME and DB_OWNER parameters are optional. If you supply both, the installer creates a new database and database owner using the names you provide. If you omit one or both of these parameters, no database or database owner are created, and the database server will only have the postgres system database. You can create additional users and databases using PostgreSQL `createdb` and `createuser` commands, or using the SQL `CREATE DATABASE` or `CREATE USER` commands.

NOTE You must type the `msiexec` command on a single line.

```
msiexec /i VMware-Postgres-9.3.5.0-0123456-x64-CIS.msi
PG_DATA_DIR="C:\ProgramData\VMware\Postgres\data"
PG_LOG_DIR="C:\ProgramData\VMware\Postgres\log" DB_OWNER=demo_user
DB_NAME=demo_db
```

Uninstall the vFabric Postgres Windows Service

You can uninstall the vFabric Postgres Windows service from a Windows physical host or virtual machine.

Prerequisites

You must have previously installed vFabric Postgres as a Windows service using the Microsoft Windows Installer (MSI). See [“Install vFabric Postgres as a Windows Service,”](#) on page 15.

Procedure

- 1 Log in as a Windows system administrator.
- 2 Uninstall the program by selecting **Start > Control Panel > Programs > Programs and Features**.
- 3 Select the vFabric Postgres Windows service, and then click **Uninstall**.

The uninstall procedure stops and removes the vFabric Postgres Windows service and the vFabric Postgres software binaries. Both the data directory and log file directory are left behind, letting you inspect or re-use them. The default locations for the data and log file directories are `C:\ProgramData\VMware\Postgres\data` and `C:\ProgramData\VMware\Postgres\log`.

Install vFabric Postgres on vCloud Hybrid Service

You can install vFabric Postgres on vCloud Hybrid Service.

When you deploy vFabric Postgres using vCloud Hybrid Service, you create a vApp template. A vApp template is a virtual machine image that is loaded with an operating system, applications, and data. These templates ensure that virtual machines are consistently configured across an entire organization.

Prerequisites

- Download the ZIP file containing the vFabric Postgres OVF for use with vCloud Hybrid Service from the VMware download site. Unzip the OVF file to your local disk drive.
- Verify that you have end user or virtual infrastructure administrator privileges. For information on vCloud Hybrid Service, see *vCloud Hybrid Service User's Guide*.

Procedure

- 1 Sign in to the vCloud Hybrid Service console.
- 2 In Virtual Machines, click **Add Virtual Machine**.
- 3 Select a virtual data center to contain the virtual machine.

The name of each available virtual data center and its available resources is displayed.

- 4 Click **Create My Virtual Machine from Scratch** at the bottom of the Select Template dialog box.

You are taken directly to the vApp Quick Access page in vCloud Director.

- 5 From the vCloud Director Home page, click **Add vApp** and follow the steps to create and configure the vApp template and its virtual machines.

To learn how to create a vApp for your vFabric Postgres deployment, see "Create a New vApp" in the *vCloud Hybrid Service User's Guide*.

- 6 When you have successfully created the vApp and shared the catalog, vCloud Hybrid Service users can create new virtual machines from the template in the My Templates section.
- 7 Power-on the vFabric Postgres virtual machine.

The first time you power-on the vFabric Postgres virtual machine, you must use the vCloud Director user interface instead of the vCloud Hybrid Service user interface. There after you can use the vCloud Hybrid Service user interface to start the vFabric Postgres virtual machine.

vFabric Postgres Client Tools and Libraries

3

You can download and install vFabric Postgres client tools for the operating system that you are using to manage and access vFabric Postgres databases. The command line front end to PostgreSQL, `psql`, is also included.

NOTE Different client tools and libraries are available for vFabric Postgres. Go to the correct download location to download the tools and libraries you need.

This chapter includes the following topics:

- [“Overview of Tools and Libraries,”](#) on page 19
- [“Client Tool Packages and Drivers,”](#) on page 20
- [“Install the Client Tools Package,”](#) on page 21
- [“Add an x86 vFabric Postgres ODBC Data Source on Windows,”](#) on page 22
- [“Relink Your Application with vFabric Postgres libpq,”](#) on page 23

Overview of Tools and Libraries

The vFabric Postgres client tools are based on the Postgres client database tools and are customized for vFabric Postgres. The tools support common configuration commands. The libraries include several APIs and the ODBC driver for PostgreSQL.

Versions for Linux x86, 32 bit and 64 bit, for Windows x86, 32 bit and 64 bit, and for Mac OS-X are available.

Linux The Linux RPM includes ODBC drivers for vFabric Postgres. The Linux ODBC driver requires unixODBC-2.3.1 or greater.

Windows The vFabric Postgres client tool installer package for Windows includes ODBC and JDBC drivers for vFabric Postgres

The vFabric Postgres client database tools that are included in the vFabric Postgres client tools packages are the same tools that are available as part of standard PostgreSQL and include `pg_dump`, `pg_restore`, and `psql`. See the postgresql.org Web site for details.

The vFabric Postgres client tools ship with the following libraries.

Table 3-1. vFabric Postgres Client Tool Libraries

Library	Description
libpq.so (Linux) or libpq.dll (Windows)	The C API to PostgreSQL. Libpq is the underlying engine for several PostgreSQL APIs such as those written for C++, Perl, Python, Tcl, and ECPG.
psqlodbcw.so (Linux) or psqlodbc35w.dll (Windows)	The ODBC driver for PostgreSQL.

The vFabric Postgres client tool libraries are customized for use with vFabric Postgres databases, but you can use the standard Postgres libraries. To ensure that you link with the vFabric Postgres libraries, do one of the following.

- If you want to keep the standard Postgres libraries on your system, ensure that your `LD_LIBRARY_PATH` environment variable specifies the location of the standard PostgreSQL libraries first.
- If you do not want to keep the standard PostgreSQL libraries, remove them and ensure that your `LD_LIBRARY_PATH` environment variable points to the location of the vFabric Postgres libraries on your system.

Client Tool Packages and Drivers

You can download client tool packages for Windows and Linux from the VMware download site. After you download the tools, you can use the drivers included in the packages.

Packages

If you plan to on writing code and on compiling an application to link with libpq, download both the client package and the development package.

You can download the client tool package for your platform from the VMware download site.

vFabric Postgres <http://vmware.com/go/download-vfabric-postgres>

You can download tools and drivers for Windows, Linux, or Mac OS X operating systems, or Java runtime environments.

Drivers

The vFabric Postgres client tools package includes a JDBC driver and an ODBC driver customized for vFabric Postgres.

JDBC Driver After installation, you can find the JDBC driver in the following locations.

Microsoft Windows	C:\Program Files\VMware\vPostgres\version\JDBC
Linux	/opt/vmware/vpostgres/current/JDBC

For example, if your application uses the JDBC driver to access a database, and you install the application as `/usr/local/lib/myapp.jar` and the PostgreSQL JDBC driver as `/usr/local/pgsql/share/java/postgresql.jar`, you run the application as follows.

```
export
CLASSPATH=/usr/local/lib/myapp.jar:/usr/local/pgsql/share/java/postgresql.jar:. java MyApp
```

ODBC Driver

The vFabric Postgres installation process installs the vFabric Postgres ODBC driver. You can verify the Windows ODBC driver installation as follows.

- 1 Select **Start > Administrative Tools > Data Sources (ODBC)**.
- 2 Click the **Drivers** tab.
- 3 Verify that the VMware vFabric Postgres ODBC driver appears in the list of installed ODBC drivers.

Install the Client Tools Package

You can install the vFabric Postgres client tools on Windows or Linux systems. The package includes drivers customized for vFabric Postgres. You can install only the base package, or install the development RPMs as well.

Prerequisites

- Prior to installing the vFabric Postgres client tools, you must install the Microsoft Visual C/C++ Runtime Library (MSVCRT). Install one or both of the following libraries depending upon whether you are running 32- or 64-bit binaries.
 - Microsoft Visual C++ 2008 Redistributable - x64 9.0.30729.6161
 - Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.6161

You can obtain these libraries from the Microsoft Download Center Web site.

- Download the vFabric Postgres client tools package for your operating system.

Procedure

- 1 Unzip the file that you downloaded and install the package.

Operating System	Installation Process
Linux	<p>Install the RPM files by running the following command.</p> <pre>rpm -ivh pathToClientRpms</pre> <pre>rpm -ivh pathToClientRpms</pre> <p><i>pathToClientRpms</i> is the full pathname of the RPM package location on your system. The default installed location is <code>/opt/vmware/vpostgres/version</code>. Use <code>-Uvh</code> instead of <code>-ivh</code> if you perform an upgrade.</p>
Windows	<ol style="list-style-type: none"> a Double-click the installer to start the installer. b Accept the license agreement and confirm the install location. <p>Installation proceeds. The default installed location is <code>\Program Files\VMware\vPostgres\version\</code>. If you install the x86 vFabric Postgres client tools on a Windows 64-bit system, the Windows installer places the client tools in <code>\Program Files (x86)\VMware\vPostgres\version\</code>.</p>
Macintosh	<p>Run or rerun the installer. You can double-click the PKG file to start the installer GUI or install from the command line by running the following command.</p> <pre># sudo installer -pkg /path/to/VMware-vPostgres-client-....pkg -target /</pre>

- 2 Ensure that your *PATH* environment variable includes the location of the vFabric Postgres client tools, for example `C:\Program Files\VMware\vPostgres\version\bin`.

What to do next

If you install both the x86 and the 64-bit vFabric Postgres client tools on a 64-bit Windows system, see [“Add an x86 vFabric Postgres ODBC Data Source on Windows,”](#) on page 22.

If you are developing a custom application, relink with `libpq`. See [“Relink Your Application with vFabric Postgres `libpq`,”](#) on page 23.

Add an x86 vFabric Postgres ODBC Data Source on Windows

If you install both the x86 and the 64-bit vFabric Postgres client tools on the same 64-bit Windows system, you must explicitly add an x86 ODBC data source.

Prerequisites

Install the x86 and the 64-bit vFabric Postgres client tools.

Procedure

- 1 In Windows Explorer, go to `C:\Windows\SysWOW64\`.
- 2 Double-click `odbcad32.exe`.
- 3 Select the **System DNS** tab and click **Add**.
- 4 Select the VMware vFabric Postgres Unicode 32-bit data source.
- 5 Click **Finish**.

Relink Your Application with vFabric Postgres libpq

If you want to use an existing PostgreSQL application with vFabric Postgres, you can relink the application.

Because vFabric Postgres `libpq.so` is dynamically linked with `libssl`, the static `ld` linker does not recognize the `rpath` of `$ORIGIN`. You can relink to specify the `rpath`.

Prerequisites

Install the vFabric Postgres client tools. You can relink without installing the development RPMs.

Procedure

- ◆ Relink with vFabric Postgres based on your operating system.

Operating System	Relinking Process						
Linux	<p>a Read /opt/vmware/vpostgres/current/share/libpq-doc/README.vpostgres-libpq.</p> <p>b Override the dynamic library search path by adding /opt/vmware/vpostgres/current/lib-public to <i>LD_LIBRARY_PATH</i>.</p> <pre># export LD_LIBRARY_PATH=/opt/vmware/vpostgres/current/lib-public # mypgapp</pre> <p>or</p> <p>c Relink using the vFabric Postgres libpq.</p> <pre># gcc -o t.t.c -L/opt/vmware/vpostgres/current/lib - WL, '-rpath=/opt/vmware/vpostgres/current/lib' -lpq</pre>						
Windows	<p>Copy libpq and other libraries to the directory of the application binaries and relink.</p> <p>By default, the libraries and header files are in the following locations.</p> <table> <tbody> <tr> <td>Development libraries</td> <td>C:\Program Files\VMware\vPostgres\version\dev</td> </tr> <tr> <td>libpgport.lib and libpq.lib libraries</td> <td>C:\Program Files\VMware\vPostgres\version\dev\lib</td> </tr> <tr> <td>libpq header files</td> <td>C:\Program Files\VMware\vPostgres\version\dev\include</td> </tr> </tbody> </table>	Development libraries	C:\Program Files\VMware\vPostgres\version\dev	libpgport.lib and libpq.lib libraries	C:\Program Files\VMware\vPostgres\version\dev\lib	libpq header files	C:\Program Files\VMware\vPostgres\version\dev\include
Development libraries	C:\Program Files\VMware\vPostgres\version\dev						
libpgport.lib and libpq.lib libraries	C:\Program Files\VMware\vPostgres\version\dev\lib						
libpq header files	C:\Program Files\VMware\vPostgres\version\dev\include						
Mac OS X	<p>Perform one of the following tasks.</p> <ul style="list-style-type: none"> ■ Override the dynamic library search path by adding the /opt/vmware/vpostgres/version/lib to the <i>DYLD_LIBRARY_PATH</i> environment variable, as follows: <pre># export DYLD_LIBRARY_PATH=/opt/vmware/vpostgres/current/lib # mypgapp</pre> ■ Relink using the vFabric Postgres libpq library during compilation. Relinking requires the Xcode developer toolset. For example, to embed the full path of libpq.dylib in the executable binary mypgapp, run this command. <pre># gcc -o mypgapp mypgapp.c - L/opt/vmware/vpostgres/current/lib -lpq</pre> ■ Relink using the vFabric Postgres libpq after compilation. Relinking requires the Xcode developer toolset. <p>NOTE This changes the binary to use vPostgres libpq.</p> <pre># install_name_tool -change "/usr/lib/libpq.5.dylib" "/opt/vmware/vpostgres/current/lib/libpq.5.dylib" mypgapp</pre> <p>To confirm which library is linked, run this command.</p> <pre># otool -L mypgapp</pre> 						

Managing vFabric Postgres

After you have installed the vFabric Postgres DBMS and the client tools, you can perform a variety of management tasks.

This chapter includes the following topics:

- [“Migrate PostgreSQL Data from Earlier Versions Into vFabric Postgres 9.3,”](#) on page 25
- [“Migrate PostgreSQL Data Into vFabric Postgres,”](#) on page 26
- [“Restarting the vFabric Postgres Service,”](#) on page 26
- [“Connection to a vFabric Postgres Database,”](#) on page 27
- [“Accounts and Services,”](#) on page 27
- [“Safeguarding Data,”](#) on page 28
- [“About vFabric Postgres Replication,”](#) on page 31
- [“Create a Replication User Account,”](#) on page 32
- [“Create a Replica Server,”](#) on page 33
- [“Promote a Replica Database to Primary Database,”](#) on page 34
- [“Monitoring Replication Status,”](#) on page 34
- [“Using Perl and Python Language Extensions,”](#) on page 35
- [“Viewing Performance Statistics,”](#) on page 36
- [“Troubleshooting Guidelines,”](#) on page 38

Migrate PostgreSQL Data from Earlier Versions Into vFabric Postgres 9.3

If you want to import PostgreSQL data or vFabric Postgres data from an earlier version, such as PostgreSQL 9.1 or vFabric Postgres 9.2, into vFabric Postgres 9.3, you can use a backup and restore process.

Procedure

- 1 Log in to the vFabric Postgres host.
- 2 Back up the existing database.

```
/opt/vmware/vpostgres/current/bin/pg_dumpall -c -h  
ip_address_of_existing_db -U postgres -l postgres
```

- 3 Restore the database on vFabric Postgres 9.3.


```
/opt/vmware/vpostgres/current/bin/psql -f
/var/vmware/vpostgres/current/mydb-backup
```
- 4 Vacuum the database after restore to make sure there is no error.


```
/opt/vmware/vpostgres/current/bin/vacuumdb -a -z -U postgres
```

Migrate PostgreSQL Data Into vFabric Postgres

You can migrate an existing PostgreSQL PGDATA directory that is using the current PostgreSQL version to vFabric Postgres by stopping the service, archiving the data, and copying the archived file.

Migrating PostgreSQL data from an earlier version of PostgreSQL requires a different process. See [“Migrate PostgreSQL Data from Earlier Versions Into vFabric Postgres 9.3,”](#) on page 25.

As part of the migration process, you stop and start the vFabric Postgres service. See [“Stop and Start the vFabric Postgres Service on the Virtual Appliance,”](#) on page 26.

Procedure

- 1 Stop the vFabric Postgres service
- 2 Archive the existing PGDATA directory using any archiving tool.
- 3 Copy the PGDATA archive to the vFabric Postgres.
- 4 Unarchive and replace the existing PGDATA directory, `/var/vmware/vpostgres/current/pgdata`.
- 5 Start the vFabric Postgres service.

Restarting the vFabric Postgres Service

You can stop and start the vFabric Postgres service on the virtual appliance or for an RPM installation. Restarting the service might be required if you change the configuration.

- [Stop and Start the vFabric Postgres Service on the Virtual Appliance](#) on page 26
Stop and start the service on the virtual appliance if you deployed the OVF template and you want to change the database configuration.
- [Stop and Start the vFabric Postgres Service for RPM Installations](#) on page 27
If you change the database configuration of your RPM installation, you can perform a configuration file reload or you can stop and start the vFabric Postgres service.

Stop and Start the vFabric Postgres Service on the Virtual Appliance

Stop and start the service on the virtual appliance if you deployed the OVF template and you want to change the database configuration.

If you deployed the vFabric Postgres OVF template, use the following commands to stop and then restart the service. For the appliance service, these commands also stop and start the VMware HA (high availability) monitor process that makes sure the database process is up and running.

Procedure

- 1 Change the configuration.
- 2 Stop the service.


```
$service vpostgres_mon stop
```

- Restart the service.

```
$service vpostgres_mon start
```

Stop and Start the vFabric Postgres Service for RPM Installations

If you change the database configuration of your RPM installation, you can perform a configuration file reload or you can stop and start the vFabric Postgres service.

If you installed the vFabric Postgres service using RPM files, the service is running within your virtual machine. In contrast to the virtual appliance, VMware HA is not integrated with the service and is not affected.

For many configuration options, a configuration file reload with `SIGHUP` or `pg_ctl reload` is sufficient.

Procedure

- Change the configuration.

- Stop the service.

```
$service vpostgres stop
```

- Restart the service.

```
$service vpostgres start
```

Connection to a vFabric Postgres Database

Connecting to a vFabric Postgres database is the same as connecting to a standard PostgreSQL database.

Accounts and Services

When you deploy the vFabric Postgres DBMS, two users are created. When the vFabric Postgres server is running, it includes services that accept remote connections.

Accounts Created During OVA Deployment

When you deploy the vFabric Postgres OVA template, the following users are created.

Table 4-1. vFabric Postgres Users Created During OVA Deployment

User Name	User Type	
root	operating system user	The root user can log into the appliance from the guest console using the same random password as the postgres user. Remote ssh logins are disabled for root. Database access is also disabled for root.
postgres	operating system user	The postgres operating system user can start the database instance and act as the super user. This user can log in to the console and into a shell.
postgres	database user	The postgres user is a database administrator account. This user can log into the appliance from the guest console, log in remotely using ssh, or connect to the database service on port 5432. The OVF deployment wizard prompts you to specify the password for all three users. Use the <code>/opt/aurora/sbin/set_password</code> command to change the password for the postgres user.

Services that Accept Remote Connections

The following vFabric Postgres services accept remote connections to the virtual appliance by default.

Table 4-2. vFabric Postgres Services that Accept Remote Connections

Service	Port
Postgres service	5432
SSH service	22
VAMI (Virtual Appliance Management Infrastructure) Web Management UI. You can connect to port 5480 via https to update or reconfigure the appliance.	5480
VAMI SFCB broker	5488 and 5489
GUI HTTPS service	5432
GUI HTTP service, which redirects to HTTPS	8080

If you are performing the RPM installation, only port 5432 is relevant.

Safeguarding Data

You create and manage data backups and recover databases using PostgreSQL Recovery Manager.

- [About Safeguarding Data](#) on page 28
Taking regular backups of your database is essential to safeguarding your data. vFabric Postgres uses PostgreSQL Recovery Manager, `pg_rman`, to perform backup and recovery of your database. You can back up your database to capture changes, preserve the database, and enable recovery, letting you restore data after a failure. You can also restore the database to its state at a particular point in time and replay changes to troubleshoot a problem.
- [Backup Types](#) on page 29
You manage backups and recover data using vFabric Postgres full backups, incremental backups, and archive recovery.
- [Schedule Database Backups](#) on page 31
You can schedule automated backups of your vFabric Postgres databases to protect your data.
- [Disable Database Backups](#) on page 31
vFabric Postgres performs database backups using the `pg_rman` utility. To temporarily halt or otherwise stop backups you disable database backups using the `crontab` utility.

About Safeguarding Data

Taking regular backups of your database is essential to safeguarding your data. vFabric Postgres uses PostgreSQL Recovery Manager, `pg_rman`, to perform backup and recovery of your database. You can back up your database to capture changes, preserve the database, and enable recovery, letting you restore data after a failure. You can also restore the database to its state at a particular point in time and replay changes to troubleshoot a problem.

vFabric Postgres provides the following features for safeguarding data.

- Backup of databases including tablespaces.
- Recovery of databases from full backups and incremental backups.
- Incremental backup and compression of backup files to optimize disk space consumption.
- Manages backup generations and can display a catalog of previous backups.

- Archive recovery.

About pg_rman for vFabric Postgres

The following features and operational details are specific to the implementation of pg_rman supplied with vFabric Postgres.

- The vFabric Postgres virtual appliance is optimized with environment variables to minimize the command strings of pg_rman. The Postgres data directory and backup directory are set for the user postgres in the .bashrc file.
- You configure pg_rman backups using the configuration file pg_rman.ini. The file is located in the directory /var/vmware/vpostgres/9.3-archive/backup/pg_rman.ini. You can manually edit the pg_rman.ini file to change your database's backup configuration settings.

The default configuration maintains two generations of full backups (KEEP_DATA_GENERATIONS=1), and seven days of data (KEEP_DATA_DAYS=7). Only backups satisfying both conditions are automatically removed.

- The pg_rman utility maintains seven days of archives (and 100 files) by default.
- Incremental and archive backups are taken at specific time intervals.
- Logs of backups are available in /var/vmware/vpostgres/log/ in the file pg_rman_\${BACKUPMODE}_\${TIMESTAMP}.log.
- The backup directory is /var/vmware/vpostgres/9.3-archive/backup. The storage space available to the backup partition is 10 GB. The WAL archive is limited to 2 GB.
- You schedule backups to run periodically at fixed times, dates, or intervals using the cron utility. The default backup plan provides a rolling, full backup once a week.

The cron files to schedule backups are located in /opt/vmware/vpostgres/current/etc. The files are pg_rman_daily.cron, pg_rman_weekly.cron and pg_rman_monthly.cron, and provide full backups daily, weekly and monthly. You can create your own customized backup plan using the default cron files as a template.

For additional information on the use of pg_rman in safeguarding your vFabric Postgres databases, refer to the open source community's documentation for PostgreSQL Recovery Manager 1.2.7.

Backup Types

You manage backups and recover data using vFabric Postgres full backups, incremental backups, and archive recovery.

Full Backups

Full backups are complete copies of the database saved to a datastore separate from the database. This section describes the pros and cons of using full backups.

Full backups use about the same amount of storage as the database itself. Because they reside on a separate disk from the database, full backups provide resiliency and benefits such as the following.

- Full backups protect against data loss due to failure of the primary data storage device.
- Full backup storage is more cost effective than using the primary data storage for backups.
- You can extend the data disk as needed.

The following are points to consider about using full backups.

- Full backups can take a long time. Large amounts of data must be copied across devices.
- Each backup uses the full size of the data disk on the backup storage device.

- The `pg_rman` utility does not support taking backups from streams using only a network connection. The backup folder must be mounted locally. However, you can take snapshots of the Virtual Machine Disk (VMDK) containing your backups and export them.

Incremental Backups

Incremental backups capture the changes to the database after the incremental backup is taken. Incremental backups initially use less storage than external backup files and take just a few minutes regardless of database size.

Incremental backups are stored in files called delta files or delta disks on the same data store as the database.

The following are points to consider about using snapshot backups.

- Because incremental backups reside on the same data store as the database, they do not protect against data loss due to failure of the data storage.
- As the database changes, the changes require more and more space on the virtual disk. That space is generally more expensive than backup storage.
- The recovery process from incremental backups is not faster than the recovery process from an external backup.

Archive Backup

If archive backup is enabled, a write-ahead log (WAL) continuously records every change made to the database while the database is running. In the event of a failure, you can replay the WAL to restore the database to its state at a point in time within the retention period of the database backups.

The WAL logs are archived and are subject to a retention period that you set. The time range for archive recovery is from the time of your oldest backup to the present. The oldest backup can be an external backup or an incremental backup.

When using the virtual appliance archiving is enabled by default. The partition dedicated to WAL archives is 2 GB of the 10 GB backup partition.

- Because every change to the database is recorded, archive backups require additional storage. Depending on how large your database is and how many transactions occur during the WAL archive retention time, the amount of storage needed can be large.

Start with a conservative storage allocation. You cannot decrease the storage allocation, but you can increase it. Monitor the size of the archive backup logs until you understand the workload and storage needed, and adjust the storage amount.

You can specify whether to suspend the database or automatically increase the log retention period if the archive backup runs out of space.

- If you generate more archive files than the archive partition can sustain (2 GB of the 10 GB backup partition by default) between two archive backups, you will not be able to recover your data (even from old snapshots) as vFabric Postgres gives priority to the newest WAL files by deleting the oldest ones when archiving WAL files from server.
- Archive backups have a performance impact on the database. The impact depends on the size of the database and the volume of database activity.
- When you enable archive backups, vFabric Postgres creates a baseline external backup.

Schedule Database Backups

You can schedule automated backups of your vFabric Postgres databases to protect your data.

You schedule vFabric Postgres backups to run periodically at fixed times, dates, or intervals using the cron utility. The cron files to schedule backups are located in `/opt/vmware/vpostgres/current/etc`. The files are `pg_rman_daily.cron`, `pg_rman_weekly.cron` and `pg_rman_monthly.cron`, and provide full backups daily, weekly, or monthly.

You can create your own customized backup plan using the default cron files as a template.

Prerequisites

You must be able to log in as the root user.

Procedure

- 1 Log in as the root user.
- 2 Run the script `/opt/vmware/vpostgres/current/etc/script_name` to specify the frequency of backups you wish to implement.

This example specifies a weekly backup of the database.

```
crontab -u postgres /opt/vmware/vpostgres/current/etc/pg_rman_weekly.cron
```

Specifying one of the above listed cron scheduling files initiates a full, rolling backup of the database daily, weekly, or monthly.

What to do next

The `pg_rman` utility is enabled by default. To temporarily halt or otherwise stop vFabric Postgres backups you must disable `pg_rman` backups using the `crontab` utility.

Disable Database Backups

vFabric Postgres performs database backups using the `pg_rman` utility. To temporarily halt or otherwise stop backups you disable database backups using the `crontab` utility.

Prerequisites

You must be able to log in as the root user.

Procedure

- 1 Log in as the root user.
- 2 Run the `crontab` command with the `-r` option. This removes the `crontab` job.

```
crontab -r postgres /opt/vmware/vpostgres/current/etc/pg_rman_weekly.cron
```

About vFabric Postgres Replication

Replication is the sharing of data between redundant resources to improve reliability, fault-tolerance, and accessibility. You can replicate a vFabric Postgres database to provide an added degree of fault tolerance and redundancy to your database environment in the event a database instance fails, or otherwise becomes unavailable.

The scripts integrate the replication features of the community version of PostgreSQL with vFabric Postgres. The scripts perform slave creation, node promotion, and replication monitoring among vFabric Postgres nodes.

- All slaves use streaming replication to catch up with master in asynchronous mode.

- If a slave is disconnected for an extended period of time from the cluster, and it cannot get the necessary WAL files from the archives of the master, you must create a new base backup. This helps maintain the archives at a low size level (a size you can customize by changing the disk size of archives in the virtual machine settings) by keeping in memory only the WAL files needed by slaves to catch up with a master.
- Recovery default is the latest time-line. This ensures that a slave trying to reconnect to a newly promoted node will catch up with the latest changes that have occurred in the cluster.
- Using cascading replication slave-to-slave connections are possible.

You will find the scripts in the directory `/opt/vmware/vpostgres/current/scripts`.

run_as_replica Transforms an existing vFabric Postgres server into a slave by synchronizing it with a given root node (either slave or master), or reconnecting to an existing node in the cluster.

show_replication_status Monitors Write-Ahead Logging (WAL) replay activity of slave nodes connected to the master node. WAL is a standard approach to transaction logging in which changes to data files are written to only after those changes have been logged. WAL allows for a significant reduction in the number of disk writes because only the log file needs to be flushed to disk at the time of transaction committal, rather than every data file changed by the transaction. Using WAL to replicate data is the fastest type of replication available, allowing you to quickly create slave instances of vFabric Postgres databases.

promote_replica_to_primary Promotes a given slave to master using the default settings in the recovery configuration (`recovery.conf`) file.

NOTE Other slave nodes can reconnect to a new master using the `run_as_replica` script.

archive_command Archives WAL files (specified with the `archive_command` in the `postgresql.conf` file, started by the vFabric Postgres server each time a WAL file is ready to be archived, or `archive_timeout` is reached). This script checks for the existence of a WAL file to be archived, and verifies the size of the archive disk to ensure that only required WAL files are maintained based on the disk space available for archives. WAL files are archived if the root node is part of a cluster as either a primary or replica server.

create_replication_user Creates a new replication user on a master node. It generates a CREATE USER query using a password that it prompts you for when you run the script.

Create a Replication User Account

You can create a database user account expressly for the purpose of replicating and promoting replica nodes.

The preferred practice when creating replica server nodes and promoting them, is to log in with a user account expressly designed for replication.

Procedure

- 1 Log in to the master database server.

- To create a replication user, run the script `/opt/vmware/vpostgres/current/scripts/create_replication_user` *replica_username*.

This example creates the replica user **replication** .

```
/opt/vmware/vpostgres/current/scripts/create_replication_user replication
```

What to do next

You can now create a replica server using the replication user account. See [“Create a Replica Server,”](#) on page 33.

Create a Replica Server

You can transform an existing vFabric Postgres server into a replica (or slave) by synchronizing it with a given root node (either slave or master), or reconnecting to an existing node in the cluster.

The `run_as_replica` script lets you create a replica of an existing root node from a read-only slave. This allows you to leverage read activity of an application with load balancing through multiple virtual machines running vFabric Postgres.

The script automatically registers the SSH authorization key of a slave node on a root node for archive transfer between nodes, so you do not need to provide additional settings for `pg_hba.conf`, `recovery.conf`, `postgresql.conf` or SSH on the root node. Note that as the `max_wal_senders` default value is set to 3, you may need to increase this value on the root node when connecting with more than one slave to a given root node.

You can also use this script to reconnect an existing slave to a new root node. In cases where the slave node is not able to catch up with the root node because of missing WAL archive files, or the slave node was in advance compared to the root node in regards to WAL replay, you must create a new base backup.

Prerequisites

Create a replication user account with which to create and promote replica server nodes. See [“Create a Replication User Account,”](#) on page 32.

Procedure

- Log in as the postgres user.
- To create a replica server, run the script `/opt/vmware/vpostgres/current/scripts/run_as_replica`. When you execute the `run_as_replica` script, specify the replication user account you created to connect to the remote node.

```
/opt/vmware/vpostgres/current/scripts/run_as_replica -h IP_MASTER -b -W -U replication
```

Option	Description
-h	Specifies the IP address of the master server.
-b	Creates a new base backup. When creating a new slave on a newly installed vFabric Postgres database, use option <code>-b</code> to create a new base backup.
-W	Specifies that the operating system prompt you to enter the password of the user you specify using the <code>-U</code> option.
-U	Specifies the user with which to connect to the remote node. The replication user is recommended.

What to do next

Once you create a replica server, you can promote it to be a primary database. See [“Promote a Replica Database to Primary Database,”](#) on page 34.

Promote a Replica Database to Primary Database

You can promote a replica (or slave) server to a primary (or master) server.

If a primary database fails, you can manually promote one of the replica database instances to take the place of the primary database.

Prerequisites

Create a replication user account with which to create and promote replica server nodes. See [“Create a Replication User Account,”](#) on page 32.

Procedure

- 1 Log in as the postgres user.
- 2 To create a replica server, run the script `/opt/vmware/vpostgres/current/scripts/promote_replica_to_primary`.


```
/opt/vmware/vpostgres/current/scripts/promote_replica_to_primary
server promoting
```

What to do next

When you promote a replica database to take over from its primary database, you must redirect other replica nodes to use the newly created primary instance. You can connect other replica nodes to a new master by running the `/opt/vmware/vpostgres/current/scripts/run_as_replica` script. See [“Create a Replica Server,”](#) on page 33.

Monitoring Replication Status

You can monitor the Write-Ahead Logging (WAL) replay activity of slave nodes connected to the master node.

WAL is a standard approach to transaction logging in which changes to data files are written to only after those changes have been logged. WAL allows for a significant reduction in the number of disk writes because only the log file needs to be flushed to disk at the time of transaction committal, rather than every data file changed by the transaction. Using WAL to replicate data is the fastest type of replication available, allowing you to quickly create slave instances of vFabric Postgres databases.

Procedure

- 1 Log in as the postgres operating system user.
- 2 Run the script `/opt/vmware/vpostgres/current/scripts/show_replication_status`.

The script monitors the WAL replay activity of slave nodes connected to the node where the script is launched.

The `replay_delta` column monitors how a given slave is late compared to a master node when replaying WAL. The lower the values of `receive_delta` and `replay_delta`, the nearer the slave node is to getting to the master in terms of WAL replay (replication state).

```
postgres@localhost:~> /opt/vmware/vpostgres/current/scripts/show_replication_status
sync_priority | slave | sync_state | log_receive_position | log_replay_position |
receive_delta | replay_delta
-----+-----+-----+-----+-----+-----
+-----+-----+-----+-----+-----+-----
0      | localhost.localdom | async | 0/80000000 | 0/80000000 | 0      | 0
(1 row)
```

Using Perl and Python Language Extensions

You can use vFabric Postgres with the PL/Perl and PL/Python language extensions. You must make sure you are using the correct versions of the language and the operating system.

PL/Python and vFabric Postgres

The PL/Python vFabric Postgres extension is supported with the following Python versions and Linux distributions.

Python version	You must install the Python 2.6.x RPM on your system. You cannot use the extension with earlier versions of Python (2.5.x) or with later versions of Python (2.7.x, 3.x).
Linux version	The RHEL 6, SLES 11 SP1, SLES 11 SP2, and SLES 11 SP3 distributions provide the Python 2.6.8 RPM and are supported for the PL/Python vFabric Postgres extension. RHEL 5.x does not provide Python 2.6.8 RPM. The PL/Python vFabric Postgres extension is not supported on RHEL 5.x.

PL/Perl and vFabric Postgres

The PL/Perl vFabric Postgres extension is supported with the following Python versions and Linux distributions.

Perl version	You must install the Perl 5.10.x RPM on your system. You cannot use the extension with earlier versions of Perl (5.8.x) or with later versions of Perl (5.12.x, 5.14.x).
Linux version	The RHEL 6, SLES 11 SP1, SLES 11 SP2, and SLES 11 SP3 distributions provide Perl 5.10.x RPMs and are supported for the PL/Perl vFabric Postgres extension. RHEL 5.x does not provide the RPMs. The PL/Perl vFabric Postgres extension is not supported on RHEL 5.x.

On supported Linux distributions, the `VMware-vPostgres-server-extensions` RPM, which contains the PL/Perl extension, includes an `install-time` scriptlet that attempts to locate the `libperl.so` shared library on the system by looking in the following locations. The scriptlet looks for the Perl binary in the following locations.

- 1 `libperl.so` path as defined by the Perl binary, where the scriptlet looks for the Perl binary in the following locations.
 - a `PATH` variable
 - b RPM location database
 - c `/usr/bin`
- 2 `libperl.so` under `/usr/lib64`
- 3 `libperl.so` under `/usr/lib`

The scriptlet creates a soft link to `libperl.so` under `/opt/vmware/vpostgres/version/lib`. If the script cannot find `libperl.so` on the system, a warning is printed during RPM installation and the PL/Perl vFabric Postgres extension might not work properly.

Viewing Performance Statistics

You can obtain vFabric Postgres performance statistics using Pgstatpack. Pgstatpack takes snapshots of various system values, and compares the differences between snapshots, letting you evaluate and tune your database's performance.

Pgstatpack lets you collect, automate, store, and view vFabric Postgres performance data. Pgstatpack stores the performance statistics permanently in database tables, which you can later use for reporting and analysis. The data collected can be analyzed using Pgstatpack reports, which include an instance health and load summary page, high resource SQL statements, and the traditional wait events and initialization parameters.

Pgstatpack creates snapshots of system information available in the system view for statistics named `pg_stat_*`, and stores the data from a snapshot in the designated table. To perform an analysis of the database's performance, the collected data from two snapshots are compared and their differences computed. You can use this information in determining appropriate performance tuning metrics for your vFabric Postgres database.

- 1 [Enable Pgstatpack](#) on page 36
You must enable the Pgstatpack extension to view performance statistics.
- 2 [Create a Pgstatpack Snapshot](#) on page 36
You create snapshots of system performance at given points in time. You then compare snapshots to evaluate system performance.
- 3 [Generating vFabric Postgres Performance Reports](#) on page 37
You generate Pgsnapstat performance reports from the snapshots you create.

Enable Pgstatpack

You must enable the Pgstatpack extension to view performance statistics.

Prerequisites

- Connect to the vFabric Postgres GUI so that you can enter and run SQL commands. See [“Enter and Run a SQL Query,”](#) on page 45.
- You must have permissions to create tables, functions, and sequences in order to use Pgstatpack.

Procedure

- ◆ To enable Pgstatpack, execute the SQL statement `CREATE EXTENSION pgstatspack;`
`CREATE EXTENSION pgstatspack;`

Create a Pgstatpack Snapshot

You create snapshots of system performance at given points in time. You then compare snapshots to evaluate system performance.

Snapshots are created by returning a result set of records from the `pgstatspack_create_snap` table.

You create snapshots at different time intervals that you compare to evaluate system performance. To manage the creation of snapshots at fixed times, dates, or intervals, use your operating system's cron utility, a time-based job scheduler. Taking snapshots at fifteen minute intervals is generally considered a good practice, but you can adjust this to be a longer or shorter interval depending on the activity of your vFabric Postgres deployment.

Prerequisites

- Connect to the vFabric Postgres GUI. See [“Access the Graphical User Interface,”](#) on page 39.
- To create a snapshot using Pgstatpack, you must first enable the Pgstatpack extension. See [“Enable Pgstatpack,”](#) on page 36.

Procedure

- 1 Connect to the vFabric Postgres GUI, and click **Enter SQL**.
- 2 Create a snapshot using the SQL statement `SELECT pgstatpack_create_snap('$DESCRIPTION');` in the Entry pane.

If you do not specify a description, the name of the user role creating the snapshot and an automatically generated timestamp is used to identify the snapshot.

This example creates a snapshot using the label `vpg_snapshot` to identify it.

```
SELECT pgstatpack_create_snap ('$vpg_snapshot');
```

What to do next

You can generate reports from the snapshots you create.

Generating vFabric Postgres Performance Reports

You generate Pgstatpack performance reports from the snapshots you create.

Performance reports are generated by comparing the differences between two snapshots using the Pgstatpack function `pgstatpack_generate_report`. You can view the list of available snapshots in the `pgstatpack_snap` table.

Prerequisites

- To generate a performance report using Pgstatpack, you must first enable the Pgstatpack extension. See [“Enable Pgstatpack,”](#) on page 36.
- You must create two or more snapshots of performance data from which to generate performance reports. See [“Generating vFabric Postgres Performance Reports,”](#) on page 37.

Procedure

- 1 Specify two snapshot IDs as arguments to the `pgstatpack_generate_report` function.

If the IDs you specify do not exist, or the start ID is newer than the stop ID, Pgstatpack returns an error.

```
SELECT pgstatpack_generate_report(1, 100);
```

The output generated by this function is a text tuple (an ordered list of elements) that you can save into another format.

- 2 You can copy the output to a directory and file that you specify in a CSV file format.

```
COPY (select pgstatpack_generate_report(1,2)) TO '/directory/filename.csv' with csv;
```

Troubleshooting Guidelines

Use the options listed in this section to analyze connection or performance problems.

Client Cannot Connect

If your client cannot connect to the vFabric Postgres appliance or to a vFabric Postgres server installed using RPMs, follow these steps to troubleshoot the issue.

- 1 Ping the server IP from your client.
- 2 Verify that Postgres is running by running the following command on the command line.

```
ps ax | grep postgres
```
- 3 Try to connect a local PostgreSQL client to the vFabric Postgres server.
- 4 Review the logs in `/var/vmware/vpostgres/current/pgdata/pg_log`.

Database Transactions Per Second Less Than Expected

If the database transactions per seconds are less than expected, follow these steps to troubleshoot the issue.

- 1 Make sure your PGDATA VMDK is on a high-performance datastore.
- 2 Look for missing indexes in your SQL queries.
- 3 Analyze concurrent queries for conflicts.
- 4 Increase the number of vCPUs and/or memory.
- 5 As a last resort, turn off `synchronous_commit` in `var/vmware/vpostgres/current/pgdata/postgresql.conf` and restart the appliance. Monitor for performance changes. See the PostgreSQL documentation for details.

Using the Graphical User Interface

A graphical user interface for managing database entities and running SQL commands is available as part of vFabric Postgres. The interface is included with the virtual appliance, and can be installed as part of RPM installation.

The tool supports database entity management and SQL management.

Database Entity Management

Database entity management includes creating, replacing, updating, and deleting database entities. These database entities include schemas, tables, views, indexes, functions, sequences, triggers, constraints, and users.

SQL Management

SQL management tasks include SQL profiling, query plan analysis, running ad-hoc queries or SQL scripts.

This chapter includes the following topics:

- [“Deploy the Graphical User Interface,”](#) on page 39
- [“Access the Graphical User Interface,”](#) on page 39
- [“Database Entity Management,”](#) on page 40
- [“SQL Management,”](#) on page 45

Deploy the Graphical User Interface

If you are using the OVA file to deploy vFabric Postgres, the GUI is included. If you are using the RPM installation, you place a WAR file on the Tomcat server to make the GUI available.

Procedure

- 1 Download the ZIP archive that has a name like `VMware-vFabric-Postgres-dbem-9.3.5.0-XXXXXX.zip` from the VMware download site and extract the `vpgdbem.war` file from the ZIP archive.
- 2 Copy the `vpgdbem.war` file to the `webapps` directory of your Tomcat server.

You can access the GUI by using the Tomcat server address and the `vpgdbem` URI. For example, if your Tomcat server is installed for port 8080, you can access the GUI at the following location:

```
http://ipaddress:8080/vpgdbem
```

Access the Graphical User Interface

The graphical user interface allows you to view database entities, create schemas and tables, and perform other management tasks. You can use the GUI console to interact with the database by using SQL.

Prerequisites

- If you are using the virtual appliance (OVA), the GUI is preinstalled.
- If you used the RPM installation process, you deploy a WAR file on your Tomcat server and access the GUI from there. See [“Deploy the Graphical User Interface,”](#) on page 39.

Procedure

- 1 Access the GUI. from the vSphere Client or by using a URL.

Option	Process
vSphere Client	<ol style="list-style-type: none"> a Use a vSphere Client to connect to the vCenter Server that manages the host or cluster on which the virtual machine runs and click the Summary tab. b If you click the Available link next to Status, you are directed to the URL of the GUI.
URL	The blue virtual appliance console that appears after all setup scripts run displays the IP address to access the GUI. The address is <i>virtual_machine_IP:8443</i> . Access the GUI from your Web browser using this address.

You cannot access the GUI from the vSphere Web Client.

- 2 When prompted, give the login credentials for the postgres database user.

Database Entity Management

Administrators perform database entity management tasks to ensure the effective and efficient operation of databases.

You can manage database entities from the **Database** tab. Managing database entities includes vacuuming and analyzing databases, and creating, altering, dropping, and browsing database entities such as the following.

- Schemas
- Tables
- Views
- Columns
- Indexes
- Sequences
- Constraints (primary, foreign, and unique key)
- Users and roles

The left pane displays schema objects. The middle pane allows you to manage individual objects.

Create a Database

The vFabric Postgres GUI allows you to create a database and customize its attributes as part of creation.

Prerequisites

Connect to the vFabric Postgres GUI. See [“Access the Graphical User Interface,”](#) on page 39.

Procedure

- 1 Right-click the host that is displayed in the left pane and select **Create Database**.

- Specify database information in the Create Database dialog.

Option	Description
Name	Name of the database (required)
Owner	Owner of the database. Always postgres for vFabric Postgres databases.
Encoding	Not currently supported.
Tablespace	Not currently supported.
Connection Limit	Number of connections. Defaults to -1, which is unlimited. Change this value only when you want to enforce that only a limited number of connections can be established with this database. Set this value lower than the maximum number of database server users. For example, if you have five databases and you want to make sure one of those databases does not have more than five users while the other databases can have as many users as the database server can handle, you use this property to limit the number of users for that database.
Comment	Comment about the purpose and characteristics of the database.

- Click **OK** to create the database.

View Database Entities

You can view and examine database entities such as schema, views, and so on from the GUI.

Procedure

- Double-click a database.
- If prompted, log in to the database
- Expand entity icons such as Catalogs and Schemas in the left pane and select individual items to view details.

What to do next

Manage database entities.

Vacuum Analyze a Database

You can use Vacuum Analyze to discover and reclaim storage occupied by dead tuples. Tuples you delete or that are made obsolete when you update the database remain in their table until you perform a Vacuum action.

You can perform Vacuum and Analyze on a database or a table.

- Use Vacuum periodically, especially on frequently updated tables, to keep the database performing well.
- Use Analyze to collect statistics about the contents of tables and to store the results.

Prerequisites

Connect to the vFabric Postgres GUI. See [“Access the Graphical User Interface,”](#) on page 39.

Procedure

- Right-click your database in the left pane and select **Vacuum Analyze Database**.

- 2 (Optional) Customize the operation.
 - To perform the Vacuum operation, uncheck the **Analyze** checkbox and click **OK**.
 - To perform the Analyze operation, uncheck the **Vacuum** checkbox and click **OK**.
 - To include the **Full** or **Freeze** operations with the Vacuum operation, check those checkboxes.
- 3 Click **OK** to perform the selected operations.

Create a Schema

After you create a database, you set up its entities, starting with the database schema.

Prerequisites

- Connect to the vFabric Postgres GUI. See [“Access the Graphical User Interface,”](#) on page 39.
- Create a database. See [“Create a Database,”](#) on page 40.

Procedure

- 1 Click the database in the left pane to display the Catalogs and Schemas icons.
- 2 Right-click **Schemas** and select **Create Schema**.
- 3 Enter the schema information.
- 4 Click **OK**.

What to do next

Create schema entities such as tables, triggers, users, and so on.

Create a Table for Schema Data

After you create a schema, you create tables to contain the schema's data.

Prerequisites

You are a database administrator or application developer setting up a database.

You created a database and a schema, and are in the **Console**.

Procedure

- 1 In the left pane, click the **Schemas** arrow to expand it.
- 2 Right-click the schema and select **Create > Table**.
- 3 Type the table name, fill factor, and comment.
- 4 Click **Next**.
- 5 Click **Add** to add a column.
 - a Type the column name, and select the column type.

Depending on the column type, you can specify a length or precision, a default value for the column, and add a comment.
 - b If users must enter a value for the column, select the **Not Null** check box.
 - c If the column is a primary key, select the **Primary Key** check box.

- 6 (Optional) In **Constraints**, select the type of constraint, **Foreign key**, **Unique**, or **Check**, that applies to the new column.

You can create foreign key constraints only if the schema has more than one table.

- a Click **Create**.
 - b Enter the conditions for the constraint, and click **OK**.
 - c Click **Next** to continue, or click **Finish** to create the table.
- 7 (Optional) In the **Auto Vacuum Settings**, select settings for removing stale data from your table,

The default settings work well for most environments. For information about autovacuum, see the documentation on the Postgres.org site.
 - 8 Click **Finish** to create the table.

Create a View

A view is a subset of related table data. For example, if you have a table that contains the locations of all corporate offices throughout the world, you can create a view of all the offices in Europe, in California, or Brazil.

Prerequisites

Verify that the table on which you want to create the view exists.

Procedure

- 1 In the left pane, click the **Schemas** arrow to expand it.
- 2 Right-click the schema and select **Create > View**.
- 3 Enter the view properties.
 - a Type a unique name in the **Name** text box.

If the name is case-sensitive, select the **Case sensitive** check box.
 - b (Optional) To restrict who can modify the view, select an owner for the view definition from the drop-down menu.
 - c Enter a SQL query to define the view.

For example, if you are creating a view of your `office_locations` table named `China Offices`, you might enter a query similar to the following to select all the office locations in China.

```
select office_name, addr1, addr2, addr3 from office_locations where country="China"
```

- 4 Click **OK**.

The view appears in the left pane under the **Views** icon.

What to do next

Examine the data in the view. See [“Examine View Data,”](#) on page 43.

Examine View Data

A view is a subset of related table data. After you create a view, you can examine the data in the view.

Prerequisites

Verify that a view is available.

Procedure

- 1 In the left pane, click the **Schemas** arrow to expand it.
- 2 Click the arrow next to the schema to expand it.
- 3 Select **Views** in the left pane.
All views under the schema appear in the list in the middle pane.
- 4 Right-click a view and select **Open**.
The view appears in the left pane.
- 5 Click the **View Data** tab to examine the data associated with the view.

Create a Constraint

Constraints let you reduce data entry errors by verifying data before inserting the data into a table.

You can create constraints when you create a table, or you can add them later by entering SQL fragments.

Prerequisites

Connect to the vFabric Postgres GUI. See [“Access the Graphical User Interface,”](#) on page 39.

Procedure

- 1 Click a table to select it, and click the gear icon.
- 2 Select **Create > Constraint**.
- 3 Select a constraint to create.

Constraint Type	Description
Check	Limits the values or value range that can be inserted in a column.
Unique	Ensures that a column or set of columns is unique.
Primary key	Uniquely identifies each row in a table. You can have only one primary key per table.
Foreign key	Points to a primary key in another table.

- 4 Complete the dialog and click **OK**.

Example: Create a Check Constraint

A check constraint evaluates to a Boolean value. Use Check constraints to determine whether a value entered for a column meets a specific truth-type requirement. For example, suppose that you create a column that must be a positive integer, such as a product price. You can create a Check constraint to return TRUE when the product price is greater than 0, and to return FALSE when the product price is less than 0. The Check constraint ensures that if a user tries to enter a negative product price, the data entry operation fails with a SQL error.

- 1 Click a table to select it.
- 2 Click the gear icon, and select **Create > Constraint**.
- 3 Select **Check Constraint**.
- 1 Type a name for the constraint, such as *check_positive_price*.
- 2 Enter the constraint in the **Check** text box.
- 3 (Optional) Enter a comment that describes the constraint.
- 4 Click **OK**.

Change the postgres Database User Password

You can change the password for the database user postgres from the console or from the vFabric Postgres GUI.

Prerequisites

Connect to the vFabric Postgres GUI. See [“Access the Graphical User Interface,”](#) on page 39.

Procedure

- 1 In the vFabric Postgres GUI, click a database arrow to display catalogs, schemas, and db login users.
- 2 Click DB Login Users.
- 3 In the right pane, right-click the postgres user and select **Properties**.
- 4 Specify a new password, confirm the password, and click **OK**.

SQL Management

Managing SQL includes developing and testing SQL queries and monitoring and tuning query performance. You must have appropriate permissions on the schema and database to develop and manage SQL queries. You can manage SQL from the schema page.

Enter and Run a SQL Query

You can enter and run SQL queries from the vFabric Postgres GUI.

Prerequisites

Connect to the vFabric Postgres GUI. See [“Access the Graphical User Interface,”](#) on page 39.

Procedure

- 1 Click **Enter SQL**.
- 2 Enter a query in the Entry pane.
 You can type or modify a SQL query, test the query, and analyze the query's execution plan before running it.
 - Type the query in the entry pane.
 - Click **Open** to open a SQL script file.
- 3 Click **Execute** to run the query.

If the query runs successfully, data appears in the Output pane.

View a Query Plan

Viewing a SQL query execution plan lets you analyze query run time and cost to ensure that your queries run as efficiently as possible.

Prerequisites

Connect to the vFabric Postgres GUI. See [“Access the Graphical User Interface,”](#) on page 39.

Procedure

- 1 Expand **Schemas** in the left pane.

- 2 Select a schema and click **Enter SQL**.
- 3 Enter a SQL query in the entry pane, or click **Open** to open a SQL script file.
- 4 Click **Execute** to run the query.
- 5 Click **Explain** to view the query plan, runtime, and CPU cost.

What to do next

Adjust the SQL query, rerun, and reexamine the query plan to tune performance.

Index

A

- accounts 27
- administer SQL 45
- analyze SQL query plan 45
- archive backup 30
- archive recovery 29
- archive_command script 31
- auto-vacuum 42

B

- backup and recovery 28
- backups
 - disable 31
 - schedule 31

C

- checkpoint 7
- client tools 19
- configure auto-vacuum 42
- constraint creation 44
- create check constraints 42
- create column constraints 42
- create constraints 44
- create database 40
- create database schema 42
- create foreign key 42
- create snapshot 36
- create SQL queries 45
- create unique constraints 42
- create views 43
- create_replication_user script 31
- cron, schedule backups 31

D

- database entities, managing 40
- database schema 42
- deploy database 8
- disable database backups 31

E

- elastic database memory 7
- enhancements 7

F

- foreign key 42

- full backup 29

G

- graphical user interface 39

I

- incremental backups 29, 30
- installation 11
- installation overview 11
- installing client tools 21

J

- JDBC 27

L

- libpq 23
- libraries 19
- linking 23
- Linux packages 20

M

- manage database entities 40
- managing databases 25
- Microsoft Windows Installer 15
- migrate 25, 26
- modify SQL queries 45

O

- ODBC data source 22
- OVA, vSphere 5 deployment 13

P

- packages 20
- password, postgres user 45
- passwords 8
- performance statistics 36
- pg_config 19
- pg_dump 19
- pg_restore 19
- pg_rman, disable 31
- pg_rman_daily.cron 31
- pg_rman_monthly.cron 31
- pg_rman_weekly.cron 31
- pgstatpack 36
- PL/Perl 35

- PL/Python **35**
- postgres user **27**
- PostgreSQL **5**
- PostgreSQL Recovery Manager **28**
- promote_replica_to_primary script **31**
- psql **19, 27**

Q

- query plan **45**

R

- relinking **23**
- remote connections **27**
- replication, Write-Ahead Logging (WAL), monitoring status **34**
- replication status **34**
- replication, about **31**
- requirements **12**
- root user **27**
- RPM, GUI **39**
- RPM Files **14**
- RPM service, stop and start **27**
- run_as_replica script **31**

S

- safeguarding data **28**
- schedule backups **31**
- schema **42**
- schema data, creating tables for **42**
- show_replication_status script **31**
- SQL management **45**
- SQL query **45**
- SQL query management **45**
- SQL query plan **45**
- start vPostgres service **26**
- stop vFabric Postgres service **26**
- system performance **36**

T

- tables, creating for schema data **42**
- troubleshooting **38**

U

- uninstall the Windows service **17**
- user interface **39**

V

- vacuum analyze a database **41**
- vCloud Hybrid Service **18**
- view data, examining **43**
- view database entities **41**
- view SQL query plan **45**

- views, creating **43**
- vpostgres_mon start **26**
- vpostgres_mon stop **26**
- vSphere 5 deployment **13**

W

- WAL archives **30**
- Windows packages **20**
- write-ahead log **30**
- Write-Ahead Logging (WAL), monitoring status **34**
- Write-Ahead Logging (WAL) replay activity **31**