

# Exporting a POSIX filesystem with GANESHA

## 1 Introduction

Thanks to its backend modules called “File System Abstraction Layers” (FSAL), GANESHA NFS server makes it possible to export any filesystem where entries can be accessed using handles.

In a POSIX filesystem, entries are accessed using their path, which does not meet the requirements for a handle (persistency, unicity). However GANESHA provides a FSAL that can assign a unic and persistent filehandle to each filesystem entry. To do this, it needs to keep some persistent information in a database.

This document will describe the database configuration needed, and the parameters GANESHA uses for accessing this database.

## 2 Database configuration

GANESHA supports both MySQL 5, PostgreSQL 7 and 8.

This section will explain how to install/configure a database in order to use the POSIX FSAL.

### 2.1 MySQL configuration

- First, install the mysql-server package
- Start the mysql server:  
`service mysqld start`
- As root, create a database for NFS-GANESHA:  
`mysqladmin create ganesha_db`
- Then, create a database user and give it all access rights.  
For this, open a SQL session (as root):  
`mysql ganesha_db`  
And execute the following commands:  
`create user 'GANESHA' identified by 'passw0rd';`  
`grant usage on ganesha_db.* to 'GANESHA'@'localhost' identified by 'passw0rd';`  
`grant all privileges on ganesha_db.* to 'GANESHA'@'localhost' identified by 'passw0rd';`  
Finally, commit the new settings:  
`flush privileges;`
- Check that this new user can access the database by opening a SQL session:  
`mysql -user=GANESHA -password=passw0rd -host=localhost ganesha_db`

- Retrieve the database schema from nfs-ganesha sources:  
`src/FSAL/FSAL_POSIX/DBExt/MYSQL/posixdb_mysql15.sql`, and execute the SQL statements it contains:  
`mysql -user=GANESHA -password=passw0rd -host=localhost ganesha_db < posixdb_mysql15.sql`
- Create a password file that will be used by NFS-GANESHA daemon:  
`echo "passw0rd" > /var/ganesha/.dbpass`  
 Don't forget setting restrictive access rights to this file:  
`chmod 600 /var/ganesha/.dbpass`

## 2.2 PostgreSQL configuration

GANESHA supports PostgreSQL version 7 and higher.

This section will explain how to install/configure a PostgreSQL 8.1 database in order to use the POSIX FSAL. For 7.x version, configuration is very similar (differences will be noticed inline).

In the following description, replace `%DBNAME%` and `%USERNAME%` signs with the actual database name and user name you want to use.

- First, install the postgresQL 8.1 package.
- Then, take the "postgres" identity (this user is created during package setup. It has all rights on PostgreSQL engine)  
`su - postgres`
- Create a new user for using the database with GANESHA:  
`createuser -no-superuser -no-createdb -no-createrole -login -pwprompt %USERNAME%` (you will be prompted for a password).

With postgresQL 7, use the following command instead:

`createuser -no-adduser -no-createdb -pwprompt %USERNAME%` (reply 'no' to questions that will be prompted, and enter a password)

- Create a new database (owned by the user we have juste created):  
`createdb -0 %USERNAME% %DBNAME%`
- In order to use PGSQL Procedural Language for improving frequent database queries, we have to activate plpgsql into our database:  
`createlang plpgsql %DBNAME%`
- Make sure you have tcp connections enabled for your database. This is set in file `'postgresql.conf'` (it should be located in `'/var/lib/pgsql/data'`). Make sure `'tcpip_socket'` parameter is true and the line is not commented:  
`tcpip_socket = true`

- In order to enable server's authentication, you have to modify `pg_hba.conf` (by default, this is located in the `/var/lib/pgsql/data` directory). At the end of the file you should have something like this:

```

local  all          all                               trust
# IPv4 local connections:
host   all          all             127.0.0.1/32      md5
host   all          all             %GANESHA_NFSD_IP%/32  md5
# IPv6 local connections:
host   all          all             ::1/128          trust

```

After this step, you have to restart the postgresql service:

```
service postgresql restart
```

- We can now create the tables in the database. To do this, retrieve the appropriate SQL script from directory `'src/FSAL/FSAL_POSIX/DBExt/PGSQL'` in GANESHA sources: use `'posixdb_v7.sql'` if you have a PostgreSQL v7.x database, `'posixdb_v8.sql'` if you are using PostgreSQL v8 database or higher version (with stored procedures support).

Then execute it like this:

```
cat posixdb_v8.sql | psql -h localhost -U %USERNAME% %DBNAME%
```

- Create a keytab file in order for GANESHA to access the database. The content of this file must have the following syntax:

```
hostname:port:database:username:password
```

Take care of setting exactly the same values in the GANESHA's configuration file for `DB_host`, `DB_port`, `DB_name` and `DB_login`.

This file's permissions MUST be 600 (rw——).

Database is now ready.

### 3 Compiling GANESHA

For using GANESHA's over a POSIX filesystem, you have to build it using the configure arg `-with-fsal=POSIX`.

Database can be selected using one of the following options `-with-db=MYSQL` or `-with-db=PGSQL`.

For PostgreSQL databases that support stored procedures (PostgreSQL PL), you can activate them with `-enable-pl-pgsql` arg.

Thus, for compiling GANESHA execute the following commands:

```

cd src
./configure --with-fsal=POSIX --with-db=MYSQL
make

```

## 4 GANESHA configuration

For configuring GANESHA's access to database, you have to set some options in the configuration file: this is done in the "POSIX" configuration block.

In this block, you must set the following values:

- `DB_host`: address of the host where the database server is running.
- `DB_name`: the database name.
- `DB_login`: user owning the database.
- `DB_keytab`: path of the keytab file.
- `DB_port`: port number where the database engine is listening on (do not set this parameter for using default).

**NB: For PostgreSQL, all those values must be exactly the same as in the database keytab file.**

Note that PostgreSQL v7 does not support alternative path for keytab file: this file must be named `pgpass` and must be located in the home directory of the user who is starting GANESHA (commonly `root`).

Here is an example of a POSIX block in the configuration file:

```
POSIX
{
    # Host
    DB_Host = "localhost";

    # Database Name
    DB_Name = "ganesha_db";

    # Login
    DB_Login = "GANESHA";

    # Path to the file where the password is stored
    # (format of the file is Database specific)
    DB_keytab = "/var/ganesha/posixdb.keytab";
}
```