FutureGrid PasswdStack Documentation

Release 0.1

Javier Diaz, Fugang Wang, Koji Tanaka, Gregor von Laszewski

August 16, 2012

CONTENTS

PasswdStack is a simple tool that allows regular users to reset their own password in the OpenStack Dashboard. Currently, OpenStack does not allow regular users to change their own password in the dashboard. This tool tackles this problem. The security is done via LDAP in the server side, which guarantees that an user can only change its own password.

FG PASSWDSTACK QUICKSTART

At this moment, our software only provides a command line interface. Thus, users need access to the machine where the FG PasswdClient client is installed. Currently, this is installed and configured in the FutureGrid India cluster (`india.futuregrid.org`).

Login on India and use the module functionality to load the environment variables:

```
$ ssh <username>@india.futuregrid.org
$ module load futuregrid
```

The authentication is done via FutureGrid Ldap server. Thus, in each command we need to specify our FutureGrid username and we will be asked for our portal password.

1.1 Using FG PasswdStack

```
$ fg-passwdstack -u <username>
```

Note: Users need to use their FutureGrid username and portal password.

After typing the command, you will be asked for your portal password. Then, you will be asked for the password that you want to have in the OpenStack Dashboard. This password is asked two times to make sure you do not misspell the new password.

```
Output:

    Passwd Stack client...
    Please insert the password for the user jdiaz
    Enter Portal Password:

    Please insert the password you want to have in the OpenStack dashboard
    Enter new Dashboard password:
    Retype new Dashboard password:
    The strength of the password is: Weak
    Connecting server: 172.29.200.121:56761
    Your request is in the queue to be processed after authentication
    Authentication OK. Your image request is being processed
    The password was reset: OK
```

FG PASSWDSTACK DOCUMENTATION

2.1 User Documentation

In this section you will find information on how to use the command line interfaces of our tools.

2.1.1 PasswdStack (fg-passwdstack)

FG PasswdStack is a service to reset the OpenStack dashboard password.

General Usage

```
usage: fg-passwdstack [-h] -u user
```

Option	Description
-h/--help	Shows help information and exit.
-u/--user <userName>	FutureGrid HPC user name, that is, the one used to login into the FG resources.

Note: These tools require users to authenticate using their FG usernames and FG portal passwords.

2.2 Administrator Documentation

In this section you will find information on software deployment details.

2.2.1 Installing FutureGrid PasswdStack

You can install the FutureGrid PasswdStack using one of the following procedures.

- *Using the "easy_install" tool*. This is the simplest option, as `easy_install` will take care of downloading and installing not just FutureGrid PasswdStack but also its dependencies. You must already have Python (version 2.6 or higher) and Python Distribute (version 0.6.15 or higher) installed on your machine (or you must be able to install them). To check if both are installed, try running `easy_install --version` from the command line. If the command is available, and it prints out a version number equal or higher than 0.6.15, you will be able to install FutureGrid PasswdStack using `easy_install`.

- *Using a source tarball*. If you are unable to install FutureGrid PasswdStack using `easy_install`, you can download a tarball with the FutureGrid PasswdStack source code. Your machine must have Python installed on it, but not the Python Distribute package (the tarball includes a setup script that will automatically download and install Python Distribute for you).

- *Downloading the latest code from GitHub*. Choose this option if you want to track the latest code in our GitHub repository,

Using `easy_install`

This option has the following prerequisites:

- Python 2.6 or higher. If Python is not available on your machine, you can find installation instructions here: http://www.python.org/getit/. Take into account that, if you are using a Linux distribution, you should be able to install it using your distribution's package manager (e.g., `apt-get install python` on Debian and Ubuntu). If you are using a Mac, Python is included by default; however, if your version is too old, take a look at the following instructions: http://www.python.org/getit/mac/

- Python Distribute 0.6.15 or higher. As noted above, you can verify if this package is installed by running `easy_install --version`. If it is not available, you can find installation instructions here: http://pypi.python.org/pypi/distribute#installation-instructions. Take into account that, although Python Distribute is included as an optional package in most Linux distributions, it is sometimes available under the name "Setuptools" (e.g., `python-setuptools` in Debian and Ubuntu systems), since Python Distribute is a fork of the Setuptools project.

If you meet these prerequisites, you should be able to install FutureGrid PasswdStack simply by running this as `root`:

```
easy_install -U futuregrid_passwdstack
```

If you are using Ubuntu or Mac OS X, you will likely just need to run this:

```
sudo easy_install -U futuregrid_passwdstack
```

If you do not have administrative privileges on your machine, you will have to install FutureGrid PasswdStack under your regular user account:

```
easy_install -U futuregrid_passwdstack --user
```

Note: Installing FutureGrid PasswdStack in your home directory will install the FutureGrid PasswdStack commands in `~/.local/bin`, which may not be in your PATH environment variable. If not, make sure to update the definition of your PATH environment variable (e.g., in the `~/.profile` file if you are using a BASH shell).

Alternatively, you can also request that the commands be installed in a directory that is already in your $PATH. You may want to use `~/bin/`, as most Linux distributions will automatically include that directory in your PATH.

```
easy_install -U futuregrid_passwdstack  --user -s ~/bin/
```

Using `pip`

`pip` is an alternative to `easy_install`. More information can be found in pip documentation.

```
pip install futuregrid_passwdstack
```

Using a source tarball

If you do not have Python Distribute, or are unable to install it, you can still install FutureGrid PasswdStack by downloading a source tarball yourself. This tarball contains an installation script that will install and setup Python Distribute, and then proceed to install FutureGrid PasswdStack.

You will first have to download the latest source tarball from the Python Package Index: http://pypi.python.org/pypi/futuregrid_passwdstack

Next, untar the tarball and run the installation script as `root`:

```
tar xvzf futuregrid_passwdstack-0.1.tar.gz
cd futuregrid_passwdstack-0.1
python setup.py install
```

Note: If you are using Ubuntu or Mac OS X, you will likely just need to run this:

```
sudo python setup.py install
```

If you do not have administrative privileges on your machine, you can choose to install everything inside your home directory:

```
python setup.py install --user
```

Tracking latest code from GitHub

If you want to use the latest version of our code from our GitHub repository, the steps are similar to installing a source tarball. However, instead of downloading a tarball, you will use git to clone our repository on your machine. Simply run the following:

```
git clone git@github.com:javidiaz/passwdstack.git
```

This will create a directory called `passwdstack`. In it, you will find the same `setup.py` script described in the previous section. If you want to install FutureGrid PasswdStack, and not make any modifications to the code, you should run `python setup.py install` as described in the previous section.

If you intend to modify the code, and want the FutureGrid PasswdStack commands to use the code in the git repository you've created on your machine, you can instead install FutureGrid PasswdStack in "developer" mode:

```
python setup.py develop
```

This will install FutureGrid PasswdStack but, instead of copying the Python source code to a system directory, it will create a pointer to the source directory you checked out. That way, any changes you make to the source code will take effect immediately (without having to reinstall FutureGrid PasswdStack).

Take into account that there are, at least, two branches in our GitHub repository: `master` and `dev`. The former always contains the latest stable release, including bug fixes, and the former contains the very latest version of our code (which may not work as reliably as the code in the `master` branch). By default, your repository will track the `master` branch. To switch to the `dev` branch, run the following:

```
git checkout dev
```

To pull the latest changes from our GitHub repository, run the following:

```
git pull origin
```

2.2.2 Setting up the FutureGrid Software for FG PasswdStack

Configuration Files

There are two places where we can locate the configuration files. Our software will look into these places in the following order:

1. In the directory `~/.fg/`

2. In the directory `/etc/futuregrid/`

If you have installed FutureGrid PasswdStack using the tarball file (*Using a source tarball*) you will find the configuration sample files in /etc/futuregrid/. Otherwise, you can download them as a tarball or a ZIP file.

Server Side: The configuration file has to be renamed as `fg-server.conf`.

Client Side: The configuration file has to be renamed as `fg-client.conf`.

Note: If you configure several clients or servers in the same machine, the `fg-client.conf` or `fg-server.conf` must be the same file.

Note: In the **Client Side**, the path of the log files must be relative to each users. Using the `$HOME` directory is a good idea.

Setting up LDAP

The authentication of our software is based on LDAP. So, we need to configure some options in the configuration files to make it possible.

Server Side

We need to configure the `[LDAP]` section. This is going to be use by all servers. More information about this section of the server configuration file can be found in *LDAP section*.

```
[LDAP]
LDAPHOST= ldap.futuregrid.org
LDAPUSER= uid=rainadmin,ou=People,dc=futuregrid,dc=org
LDAPPASS= passwordrainadmin
log= ~/fg-auth.log
```

Setting up FG PasswdStack

In this section we explain how to configure FG PasswdStack.

Server Side

First, we are going to configure the main server. We need to configure the `[PasswdStackServer]` Section (see *PasswdStackServer section*).

```
[PasswdStackServer]
port = 56795
proc_max=5
refresh=20
log = passwdstackserver.log
log_level = debug
ca_cert=/opt/futuregrid/futuregrid/etc/imdserver/cacert.pem
certfile=/opt/futuregrid/futuregrid/etc/imdserver/imdscert.pem
keyfile=/opt/futuregrid/futuregrid/etc/imdserver/privkey.pem
```

Once everything is set up, you can start the server executing `PasswdStackServer.py` as `imageman` user.

Note: We recommend to have a system user that run all the servers (i.e. imageman). In this way, it will be easier to manage the sudoers file when necessary.

Client Side

In the client side, we need to configure the `[PasswdStack]` section. More information about this section of the client configuration file can be found in *PasswdStack section*.

```
[PasswdStack]
port = 56796
serveraddr=123.123.123.123
log = passwdstackclient.log
log_level = debug
ca_cert=/etc/futuregrid/imdserver/cacert.pem
certfile=/etc/futuregrid/imdserver/imdscert.pem
keyfile=/etc/futuregrid/imdserver/privkey.pem
```

The executable file of this client is `fg-paswdstack`. More information about how to use FG PasswdStack can be found in the *FG PasswdStack Manual*.

FG PasswdStack Check List

	Server Side (`fg-server.conf`)	Client Side (`fg-client.conf`)
Requirement	• LDAP client configured in the machine	
Configure	• `[PasswdStackServer]` section • `[LDAP]` section	• `[PasswdStack]` section
Executables	• `PasswdStackServer.py`	• `fg-paswdstack`

2.2.3 Configuration file reference

`fg-server.conf` configuration file

Section `[LDAP]`

This section is used to configure the access to LDAP to verify the user passwords.

This section is required by all services

Option `LDAPHOST`   **Type:** String

Required: Yes

Hostname or IP address of the LDAP server that manages the user's authentication.

Option `LDAPUSER`   **Type:** user-dn

Required: Yes

This is the DN of an user that have read access to the encrypted passwords of every user. This looks like `uid=USER,ou=People,dc=futuregrid,dc=org`

Option `LDAPPASS`   **Type:** String

Required: Yes

Password of the user specified in the previous section.

Option `log`   **Type:** log-file

Required: Yes

Location of the file where the logs will be stored.

Option `test`   **Valid values:** `True, False`

Required: No

This option is for development purposes. For security reasons, the LDAP server cannot be contacted from outside of FutureGrid network. Therefore, we need this option to go test our services before we deploy them on production.

Section `[PasswdStackServer]`

This section is used to configure the FG Move Server.

Option `port`   **Type:** Integer

Required: Yes

Port where the FG PasswdStack server will be listening.

Option `proc_max`   Type: Integer

Required: Yes

Maximum number of request that can be processed at the same time.

Option `refresh`   Type: Integer

Required: Yes

Interval to check the status of the running requests when `proc_max` is reached and determine if new request can be processed.

Option `log`   Type: log-file

Required: Yes

Location of the file where the logs will be stored.

Option `log_level`   Valid values: `debug`, `error`, `warning`, `info`

Required: No

Desired log level. The default option is `debug`.

Option `ca_cert`   Type: ca-cert

Required: Yes

Location of CA certificate (PEM-encoded) used to generate user and service certificates. Server certificates.

Option `certfile`   Type: service-cert

Required: Yes

Location of the certificate (PEM-encoded) used by the FG PasswdStack server. Server certificates.

Option `keyfile`   Type: key-cert

Required: Yes

Location of the private key (PEM-encoded) of the certificate specified in `certfile`. Server certificates.

`fg-client.conf` configuration file

Section `[PasswdStack]`

This section is used to configure FG PasswdStack.

Option `port`   Type: Integer

Required: Yes

Port where the FG PasswdStack server will be listening.

Option `serveraddr` **Type:** String

Required: Yes

Address of the machine where the FG PasswdStack server is running.

Option `log` **Type:** log-file

Required: Yes

Location of the file where the logs will be stored.

Option `log_level` **Valid values:** debug,``error``,``warning``,``info``

Required: No

Desired log level. The default option is debug.

Option `ca_cert` **Type:** ca-cert

Required: Yes

Location of CA certificate (PEM-encoded) used to generate user and service certificates.

Option `certfile` **Type:** service-cert

Required: Yes

Location of the certificate (PEM-encoded) used by the FG PasswdStack client.

Option `keyfile` **Type:** key-cert

Required: Yes

Location of the private key (PEM-encoded) of the certificate specified in certfile.

2.2.4 Configuring a Module for FutureGrid Software

The software packages on the FutureGrid machines is manage using the Environment Modules. The Environment Modules package provides for the dynamic modification of a user's environment via modulefiles.

In this section, we explain how to create a module for our software.

1. Create a directory to place the software /N/soft/futuregrid-1.0/.

2. Locate the directory where Modules is installed. In the case of India, this is installed in /opt/Modules/. From now on we will refer to this location as $MODULES_PATH.

3. Create a directory in $MODULES_PATH/default/modulefiles/tools/futuregrid

4. In this directory we need to create a file with the version number. In this example the file is named 1.0. The content of this file is some information about the software location and the list of modules that need to be loaded as requirements.

```
#%Module1.0#######################################################

set ver 1.0
set path /N/soft/futuregrid-$ver

proc ModulesHelp { } {
puts stderr "This module adds the FutureGrid toolkit to your environment"
}

module-whatis "Configures your environment for the FutureGrid toolkit"

prepend-path PATH $path
prepend-path PATH $path/bin/

if [ module-info mode load ] {
puts stderr "futuregrid version $ver loaded"
}

if [ module-info mode switch2 ] {
puts stderr "futuregrid version $ver loaded"
}

if [ module-info mode remove ] {
puts stderr "futuregrid version $ver unloaded"
}

module load euca2ools
module load python_w-cmd2
module load moab
module load torque
```

Note: If the python is not the one installed in the system, the binaries may be inside your python directory.

1. In case the software binaries were copied into `/usr/bin` or `/usr/local/bin`. We need to move them to the directory `/N/soft/futuregrid-1.0/bin/`

2.2.5 Building the FG PasswdStack Documentation

The documentation has been created using Sphinx and therefore you have to build it before you can see the final html files. The source files of the documentation can be found under the `doc` directory of our software package. Next, we define the needed steps to build the documentation.

1. Install the documentation *Using a source tarball* or *Downloading the latest code from GitHub*.

2. Change your current directory to the `doc` one.

   ```
   cd doc
   ```

3. Build the documentation using the Makefile.

   ```
   make website
   ```

4. The documentation should be in the directory `build/web-<version>/`. This basically contains html files that can be copied to a regular http server.

2.3 Others

2.3.1 Changelog and Release Notes

0.1

Summary of the main functionality provided for this version:

- **Security**

    - Authentication is performed using the FutureGrid LDAP server

- Change the OpenStack password for an user by executing the command keystone user-password-update

DOWNLOAD

The latest version of the FutureGrid PasswdStack is 0.1

The recommended way of installing Move is using `easy_install`. You can find instructions on how to do this in the *Installation* chapter of the documentation.

If you do need to download FG Move as an installable Python Egg or as a source tarball, both are available for download in the Python Package Index: http://pypi.python.org/pypi/futuregrid_passwdstack

3.1 Sample files

The sample configuration files shown in the documentation are available for download as a tarball or a ZIP file.

3.2 Development version

If you want to use the latest development version of FG Move, you can track our GitHub repository.

The *Installation* chapter of the documentation provides instructions on how to check out our code from GitHub.

SUPPORT

If you run into problems when using FutureGrid PasswordStack, please use our help form at https://portal.futuregrid.org/help

KNOWN PROBLEMS

Depending of the configuration of your machines, you may have problems installing some python modules. In particular, we have notice problems with the ldap modules, which is required for the installation of our software.

- Installing python-ldap

```
Ldap api
sudo apt-get install python-ldap
or
sudo yum install python-ldap
or
sudo yum install python26-ldap  #(when the default python is the 2.4 version)
```