



# **FutureGrid Move Documentation**

*Release 0.1*

**Javier Diaz, Fugang Wang, Gregor von Laszewski**

September 11, 2012



# CONTENTS

<b>1</b>	<b>What is FutureGrid Move?</b>	<b>3</b>
1.1	Summary . . . . .	3
1.2	Use Cases . . . . .	4
<b>2</b>	<b>FG Move QuickStart</b>	<b>5</b>
2.1	Requirements . . . . .	5
2.2	Using FG Move . . . . .	5
<b>3</b>	<b>FG Move Documentation</b>	<b>7</b>
3.1	User Documentation . . . . .	7
3.2	Administrator Documentation . . . . .	10
3.3	Others . . . . .	24
<b>4</b>	<b>Download</b>	<b>25</b>
4.1	Sample files . . . . .	25
4.2	Development version . . . . .	25
<b>5</b>	<b>Support</b>	<b>27</b>
<b>6</b>	<b>Known problems</b>	<b>29</b>



FutureGrid Move is a service that enables physical resources re-allocation among infrastructures. By using a simple command line interface, this service is able to de-register a machine from a particular infrastructure and register it in another one. Internally, this service makes use of [Teefaa](#) to dynamically provision the selected machine with the OS and software needed for a successful registration in the new infrastructure. FG Move also maintains a database with information about the machines composing each one of the different infrastructures. The database can be consulted to obtain detailed information about a particular infrastructure.

Currently, we support HPC infrastructures (Torque-based) and cloud infrastructures based on [Eucalyptus](#) and [Open-Stack](#) frameworks. Additionally, we are working to provide support to other cloud infrastructures based on [Nimbus](#) and [OpenNebula](#) frameworks.



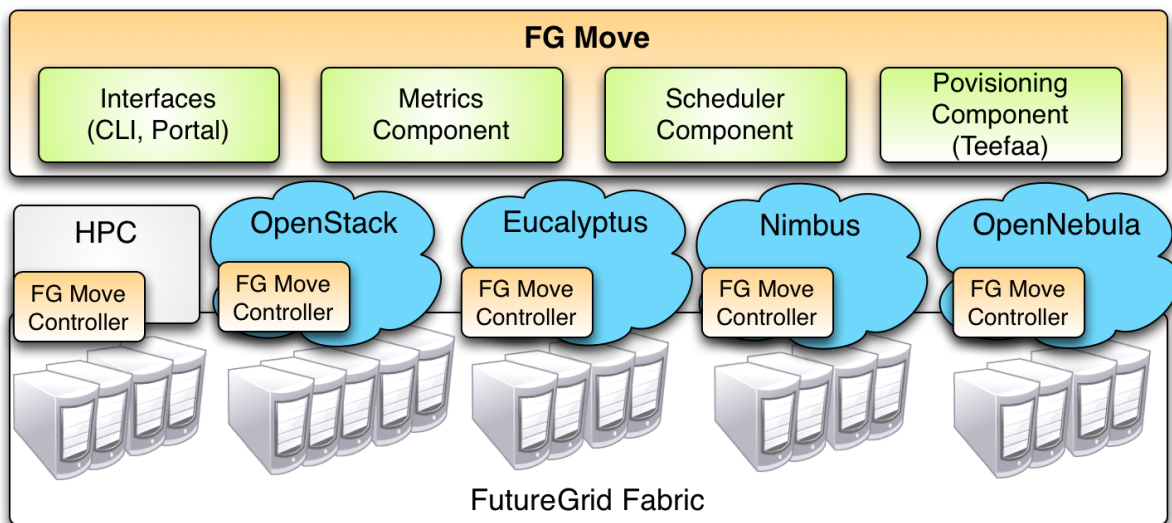
# WHAT IS FUTUREGRID MOVE?

## 1.1 Summary

FutureGrid Move is a service that enables physical resources re-allocation among infrastructures. By using a simple command line interface, this service is able to de-register a machine from a particular infrastructure and register it in another one. Internally, this service makes use of [Teefaa](#) to dynamically provision the selected machine with the OS and software needed for a successful registration in the new infrastructure. FG Move also maintains a database with information about the machines composing each one of the different infrastructures. The database can be consulted to obtain detailed information about a particular infrastructure.

An scheduler will be created to integrate FG Move with our metric [FutureGrid Metrics](#). In this way, it will allow us to make reservations or analyze the historical usage to automatically move resources according to user demand.

The following picture represents the architecture of FG Move.



The workflow that “moves” a machine from one infrastructure to another is the following:

- The client requests to move a resource.
- **FG Move Server contacts with the FG Move Controller of the original infrastructure to de-register the machine.**
  - If the machine is idle, it is de-registered.

- If the machine is not idle, it waits for the machine to become idle (for a certain amount of time) or it terminates all its tasks/VMs. Then, the machine is de-registered.
- FG Move Server calls Teefaa to provision the machine with the appropriated OS. Teefaa will repartition the disk of the machine and place the required OS.
- FG Move Server contacts with the FG Move Controller of the destination infrastructure to register the machine.

Currently, we support HPC infrastructures (Torque-based) and cloud infrastructures based on [Eucalyptus](#) and [OpenStack](#) frameworks. Additionally, we are working to provide support to other cloud infrastructures based on [Nimbus](#) and [OpenNebula](#) frameworks.

## 1.2 Use Cases

- Re-allocate machines based on user demands. This can be initiated by creating a reservation, by predicting future needs (learning for historical information), or manually.
- Conduct tests on the same machines with different frameworks.
- Re-use images created for experiments.



# FG MOVE QUICKSTART

## 2.1 Requirements

At this moment, our software only provides a command line interface. Thus, users need access to the machine where the FG Move 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
```

---

**Note:** FG Move is only available for **administrators**.

---

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. After the user is successfully authenticated against the Ldap server, it is verified if that user is authorized to use the FG Move service.

```
$ fg-move -u <username> ...
```

---

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

---

## 2.2 Using FG Move

The command line of this service is organized in three specialized subparsers.

- **node.** A node is the representation of a physical machine. This subparser exposes the functionality to operate with them.
- **cluster.** A cluster is a set of nodes identified by a name (representation of a physical cluster). This subparser exposes the functionality to operate with them.
- **service.** A service is the representation of an infrastructure, which is composed by nodes. It exposes the functionality to move physical machines from one infrastructure to another.

### 2.2.1 Operating with Clusters

- Create a cluster named Indiana

```
fg-move -u jdiaz cluster --create Indiana
```

- List available clusters

```
fg-move -u jdiaz cluster --list
```

- List information about a particular cluster

```
fg-move -u jdiaz cluster --list Indiana
```

## 2.2.2 Operating with Nodes

- Add a node to a particular cluster. We need to indicate the properties of the node, which are a node ID, hostname, IP and cluster ID.

```
fg-move -u jdiaz node --add i90.india i90 10.0.1.2 Indiana
```

- List information of a particular node.

```
fg-move -u jdiaz node --info i90.india
```

## 2.2.3 Operating with Services

- Create a service named IndianaOpenStack.

```
fg-move -u jdiaz service --create IndianaOpenStack
```

- List available services.

```
fg-move -u jdiaz service --list
```

- List information about a particular service.

```
fg-move -u jdiaz service --list IndianaOpenStack
```

- List nodes that are not allocated into any service.

```
fg-move -u jdiaz service --listfreenodes
```

- Add nodes to the service IndianaOpenStack. The nodes must be free, that is they cannot be allocated to another service.

```
fg-move -u jdiaz service --add i90.india i20.india IndianaOpenStack
```

- Remove a node from a service. The node must be idle to perform this operation or you have to force the operation by adding `--force`.

```
fg-move -u jdiaz service --remove i20.india IndianaOpenStack
```

- Move a node from a service to another. The node must be idle to perform this operation or you have to force the operation by adding `--force`.

```
fg-move -u jdiaz service --move i90.india IndianaOpenStack IndianaEucalyptus
```

# FG MOVE DOCUMENTATION

## 3.1 User Documentation

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

### 3.1.1 Move (fg-move)

FG Move is a service to re-allocate resources among infrastructures.

#### General Usage

```
usage: fg-move [-h] -u user {node,cluster,service} ...
```

positional arguments:

{node,cluster,service} Positional arguments group different options that can be displayed by specifying <positional\_argument> -h

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.

#### Node subparser

```
usage: fg-move -u user node [-h] (-a nodeId hostname ip cluster | -r nodeId cluster | -i nodeId ) [-f]
```

Options between brackets are not required. Parenthesis means that you need to specify one of the options.

Option	Description
-h/--help	Shows help information and exit.
-a <nodeId> <hostname> <ip> <clusterId>, --add <nodeId> <hostname> <ip> <clusterId>	Add new node to a cluster.
-r <nodeId> <clusterId>, --remove <nodeId> <clusterId>	Remove node. It is also removed from cluster and service. (not supported yet)
-i <nodeId>, --info <nodeId>	Information of a node.
-f, --force	Force operation.

### Examples

- Add a node to a particular cluster. We need to indicate the properties of the node, which are a node ID, hostname, IP and cluster ID.

```
fg-move -u jdiaz node --add i90.india i90 10.0.1.2 Indiana
```

- List information of a particular node.

```
fg-move -u jdiaz node --info i90.india
```

### Cluster subparser

```
usage: fg-move -u user cluster [-h] (-c clusterId | -r clusterId | -l [clusterId]) [-f]
```

Options between brackets are not required. Parenthesis means that you need to specify one of the options.

Option	Description
-h/--help	Shows help information and exit.
-c <clusterId>, --create <clusterId>	Create a new cluster.
-r <clusterId>, --remove <clusterId>	Remove cluster (not supported yet)
-l [clusterId], --list [clusterId]	List available clusters or the information about a particular one if a clusterId is provided.
-f, --force	Force operation.

### Examples

- Create a cluster named Indiana

```
fg-move -u jdiaz cluster --create Indiana
```

- List available clusters

```
fg-move -u jdiaz cluster --list
```

- List information about a particular cluster

```
fg-move -u jdiaz cluster --list Indiana
```

### Service subparser

```
usage: fg-move -u user service [-h] (-c serviceId type | -a nodeId/s... [nodeId/s...] serviceId | -r nodeId/s... [nodeId/s...] serviceId | -m nodeId/s... [nodeId/s...] serviceIdorigin serviceIddestination | -l [clusterId] [-f])
```

Options between brackets are not required. Parenthesis means that you need to specify one of the options.

Option	Description
-h/--help	Shows help information and exit.
-c/--create <serviceId>	Create a new service.
-a/--add <nodeId/s> [nodeId/s...] <serviceId>	Add a node or list of nodes to a service.
-r/--remove <nodeId/s> [nodeId/s...] <serviceId>	Remove a node or list of nodes from a service.
-m/--move <nodeId/s> [nodeId/s...] <serviceIdorigin> <serviceIddestination>	Move a node or list of nodes from a service to another.
-i/--info <nodeId/s> [nodeId/s...] <serviceId>	Information about the status of node/s from a service (idle, busy, does not exists).
-l/--list [serviceId]	List available services or the information about a particular one if a clusterId is provided.
-s/--listfreenodes [clusterId]	List of nodes that are not assigned to any service. The list can be restricted to a particular cluster
-f/--force	Force operation.

## Examples

- Create a service named IndianaOpenStack.

```
fg-move -u jdiaz service --create IndianaOpenStack
```

- List available services.

```
fg-move -u jdiaz service --list
```

- List information about a particular service.

```
fg-move -u jdiaz service --list IndianaOpenStack
```

- List nodes that are not allocated into any service.

```
fg-move -u jdiaz service --listfreenodes
```

- Add nodes to the service IndianaOpenStack. The nodes must be free, that is they cannot be allocated to another service.

```
fg-move -u jdiaz service --add i90.india i20.india IndianaOpenStack
```

- Remove a node from a service. The node must be idle to perform this operation or you have to force the operation by adding --force.

```
fg-move -u jdiaz service --remove i20.india IndianaOpenStack
```

- Move a node from a service to another. The node must be idle to perform this operation or you have to force the operation by adding --force.

```
fg-move -u jdiaz service --move i90.india IndianaOpenStack IndianaEucalyptus
```

---

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

---

## 3.2 Administrator Documentation

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

### 3.2.1 Installing FutureGrid Move

You can install the FutureGrid Move 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 Move 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 Move using `easy_install`.
- *Using a source tarball.* If you are unable to install FutureGrid Move using `easy_install`, you can download a tarball with the FutureGrid Move 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 Move simply by running this as `root`:

```
easy_install -U futuregrid_move
```

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

```
sudo easy_install -U futuregrid_move
```

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

```
easy_install -U futuregrid_move --user
```

---

**Note:** Installing FutureGrid Move in your home directory will install the FutureGrid Move 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_move --user -s ~/bin/
```

---

## Using pip

pip is an alternative to easy\_install. More information can be found in [pip documentation](#).

```
pip install futuregrid_move
```

## Using a source tarball

If you do not have Python Distribute, or are unable to install it, you can still install FutureGrid Move 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 Move.

You will first have to download the latest source tarball from the Python Package Index: [http://pypi.python.org/pypi/futuregrid\\_move](http://pypi.python.org/pypi/futuregrid_move)

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

```
tar xvzf futuregrid_move-0.1.tar.gz
cd futuregrid_move-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:futuregrid/rain-move.git
```

This will create a directory called rain-move. In it, you will find the same setup.py script described in the previous section. If you want to install FutureGrid Move, 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 Move commands to use the code in the git repository you've created on your machine, you can instead install FutureGrid Move in “developer” mode:

```
python setup.py develop
```

This will install FutureGrid Move 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 Move).

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
```

### 3.2.2 Setting up the FutureGrid Software for FG Move

#### 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 Move 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 Move

In this section we explain how to configure FG Move. FG Move is composed by a main server that orchestrates different FG Move controllers distributed among the different infrastructures head nodes. These controllers make sure that the machines are properly added or removed from the infrastructure.

### Server Side

First, we are going to configure the main server. We need to configure the [RainMoveServer] Section (see *MoveServer section*) in the `fg-server.conf` configuration file.

```
[RainMoveServer]
port = 56795
proc_max=5
refresh=20
authorizedusers = adminuser1, adminuser2
protectedmachines = im1, 172.29.22.1, i50, 172.29.20.50, i135, 172.29.20.135, i136, 172.29.20.13
log = moveserver.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
Clientca_cert=/opt/futuregrid/futuregrid/etc/imdclient/cacert.pem
Clientcertfile=/opt/futuregrid/futuregrid/etc/imdclient/imdccert.pem
Clientkeyfile=/opt/futuregrid/futuregrid/etc/imdclient/privkey.pem
dbaddress = localhost
dbport = 23000
dbname = fg_move
```

Next we need to create sections for each one of the FG Move controllers (see *Move Remote Site Controllers*).

```
[Move-eucalyptus-indiaeuca]
address=129.79.49.12
port=56804

[Move-openstack-indiaopenstack]
address=129.79.49.13
port=56805

[Move-hpc-indiahpc]
address=129.79.49.10
port=56806

[Move-nimbus-hotelnimbus]
address=129.79.49.11
port=56807
```

---

**Note:** Make sure that `port` and `address` matches with those used by the Move controllers (see *Move Site Server*).

---

We can also create an inventory file to initialize our database. This file will describe the nodes, clusters and services. This file will be read by the RainMoveServer during its initialization (first time it is executed). The inventory

information will be automatically stored in a MongoDB database. Therefore, this inventory file will not be needed anymore.

In the inventory file, the nodes are defined by its Id, hostname and IP. They have to be part of a cluster, which is defined by the **CLUSTER:** label followed by the name of the cluster. We also have services that contain the Id of the nodes that are allocated to each service. Services are defined with the **SERVICE** label, the type of service (**OPENSTACK**, **NIMBUS**, **EUCALYPTUS**, **HPC**).

This file looks like:

```
[CLUSTER:HOTEL]
c01.hotel,c01,149.165.147.1
c02.hotel,c02,149.165.147.2

[CLUSTER:INDIA]
i01.india,i01,149.165.148.1
i02.india,i02,149.165.148.2
i100.india,i200,149.165.148.100
i101.india,i101,149.165.148.101
i102.india,i102,149.165.148.102

[SERVICE:NIMBUS:HOTELNIMBUS]
c01.hotel
c02.hotel

[SERVICE:EUCALYPTUS:INDIAEUCA]
i100.india
i102.india

[SERVICE:OPENSTACK:INDIAOPENSTACK]
i101.india

[SERVICE:HPC:INDIAHPC]
i01.india
i02.india
```

### Requirements

- Install [Teefaa](#) in the same machine (see [Teefaa Documentation](#)).
- Install MongoDB and pymongo on the machine (see [MongoDB Deployment](#)).

Once everything is set up (including Teefaa, MongoDB and pymongo), you can start the fg-move server as imageman user.

- The first time you start your service you need to provide the inventory file to initialize the database. Thus, you execute the service using `RainMoveServer.py -l <inventoryfile>`. This will delete the database defined in the parameter `dbname` defined in the `fg-server.conf` configuration file under the section `[RainMoveServer]`, if it exists.
- The following times, you need to start the fg-move service by executing `RainMoveServer.py`.

---

**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 `[RainMove]` section in the `fg-client.conf` configuration file. More information about this section of the client configuration file can be found in [Move section](#).

```
[RainMove]
port = 56795
serveraddr=localhost
log=~/.clientrainmove.log
log_level=debug
ca_cert=/etc/futuregrid/imdclient/cacert.pem
certfile=/etc/futuregrid/imdclient/imdccert.pem
keyfile=/etc/futuregrid/imdclient/privkey.pem
```

Once you have everything set up, any user that is in the `authorizedusers` field of the section `[RainMoveServer]` will be able to use this service (see [Rain Move Server](#)). Of course, the user has to authenticate against LDAP too.

The executable file of this client is `fg-move`. More information about how to use FG Move can be found in the [FG Move Manual](#).

## FG Move Check List

	Server Side ( <code>fg-server.conf</code> )	Client Side ( <code>fg-client.conf</code> )
<b>Requirement</b>	<ul style="list-style-type: none"> <li>• Teefaa installed and configured in the same machine</li> <li>• MongoDB and Pymongo installed and configured in the same machine</li> </ul>	
<b>Configure</b>	<ul style="list-style-type: none"> <li>• <code>[RainMoveServer]</code> section</li> <li>• <code>[LDAP]</code> section</li> <li>• Move Site controllers sections <code>Move-&lt;service&gt;-&lt;serviceID&gt;</code></li> </ul>	<ul style="list-style-type: none"> <li>• <code>[RainMove]</code> section</li> </ul>
<b>Executables</b>	<ul style="list-style-type: none"> <li>• <code>RainMoveServer.py</code></li> </ul>	<ul style="list-style-type: none"> <li>• <code>fg-move</code></li> </ul>

## Setting up FG Move Site Controller

In this section, we explain how to configure the FG Move Site Controller. As we said previously, this service will make sure that the machines are properly added or removed from the infrastructure. Therefore, it has to run in the machine that controls the infrastructure (i.e. where Torque, OpenStack Nova manager or Eucalyptus cloud controller is installed). Our service controller is generic and works for every supported infrastructure, we only need to tune it up properly.

## Server Side

In the Server side we need to configure the `[RainMoveSiteServer]` Section (see [RainMoveSiteServer section](#)). Here, we have an example for a Cloud service (we have defined the `ec2varfile` option). In the case of HPC, we

need to define the option `hpcproperties` instead (i.e. `hpcproperties = compute`).

```
[RainMoveSiteServer]
port = 56796
proc_max = 5
refresh = 20
log = movesiteserver.log
log_level = debug
max_wait = 1000
ec2varfile = ~/eucarc
ca_cert=/etc/futuregrid/imdserver/cacert.pem
certfile=/etc/futuregrid/imdserver/imdscert.pem
keyfile=/etc/futuregrid/imdserver/privkey.pem
```

---

**Note:** Make sure that the `port` specified here matches with the one specified in the sections defined when configuring the main *FG Move server*.

---

Next, we need to do specific configurations depending on the infrastructure we are targeting. Typically, you only need to do one of the following subsections per management machine.

- **HPC infrastructure**

Considering that `imageman` is the user that will run the service, we need to allow this user to execute the `pbsnodes` command and some other commands as root user.

Edit `sudoers` file by executing `visudo` as root user and add the following lines:

```
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin:/opt/moab/bin:/opt/torque/bin

imageman ALL=(ALL) NOPASSWD: /opt/moab/bin/mschedctl -R
imageman ALL=(ALL) NOPASSWD: /opt/torque/bin/pbsnodes
imageman ALL=(ALL) NOPASSWD: /opt/torque/bin/qmgr
imageman ALL=(ALL) NOPASSWD: /opt/torque/bin/qdel
```

- **Eucalyptus Infrastructure**

Here, we need to provide `imageman` with an Eucalyptus credentials of an admin user. This is basically needed to terminate instances when we use the `--force` option with `fg-move`. A good place to place the credentials is the home directory of this user.

Next, we need to edit `sudoers` file by executing `visudo` as root user and add the following lines:

```
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin:/opt/euca/bin

imageman ALL=(ALL) NOPASSWD: /opt/euca/bin/euca_conf
```

- **OpenStack Infrastructure**

Here, we need to provide `imageman` with an OpenStack credentials of an admin user. This is basically needed to terminate instances when we use the `--force` option with `fg-move`. A good place to place the credentials is the home directory of this user.

Next, we need to edit `sudoers` file by executing `visudo` as root user and add the following lines:

```
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin:/opt/openstack/bin

imageman ALL=(ALL) NOPASSWD: /opt/openstack/bin/nova-manage
```

- **Nimbus Infrastructure**

Not developed yet

- **OpenNebula Infrastructure**

Not developed yet

Once everything is set up, you can start the server by execution `RainMoveSiteServer.py` as `imageman` user.

### FG Move Site Check List

	Server Side ( <code>fg-server.conf</code> )
<b>Access to</b>	<ul style="list-style-type: none"> <li>• Cloud credentials of an admin user (not needed for HPC)</li> </ul>
<b>Configure</b>	<ul style="list-style-type: none"> <li>• <code>[RainMoveSiteServer]</code> section</li> <li>• <code>sudoers</code></li> </ul>
<b>Executables</b>	<ul style="list-style-type: none"> <li>• <code>RainMoveSiteServer.py</code></li> </ul>

### 3.2.3 Deployment of MongoDB

In this section we are going to explain how to install a single MongoDB service. More information can be found in [MongoDB](#)

**Note:** In MongoDB the databases and tables are created automatically.

Remember that MongoDB is case sensitive.

#### Install MongoDB

- Download the package file for you specific distribution. They can be found in <http://www.mongodb.org/downloads>.

- Uncompress the file.

```
tar cvfz <filename>
```

- Create DB directory

```
sudo mkdir -p /data/db/
sudo chown `id -u` /data/db      #The owner must be the user that is going to execute mongod
```

- Run MongoDB

```
mongod --port 23000 --dbpath /data/db/ --fork --logpath=/data/db/mongo.log
```

- Test MongoDB using the client.

```
$ mongo --port 23000

> db.foo.save ( {a:1} )
> db.foo.find ()
```

## Install pymongo

The latest documentation can be found in <http://api.mongodb.org/python/2.3rc1/installation.html>.

- Install with pip

```
pip install pymongo
```

- Install with easy\_install

```
easy_install install pymongo
```

## 3.2.4 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 [RainMoveServer]**

This section is used to configure the FG Move Server.

**Option port**    **Type:** Integer

**Required:** Yes

Port where the FG Move 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 authorizedusers**    **Type:** String-list (comma separated)

**Required:** No

List of users (separated by commas) that can use this service.

**Option protectedmachines**    **Type:** String-list (comma separated)

**Required:** No

List of machines names, identifiers or IPs (separated by commas) that are protected and cannot be reinstalled.

**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 Move 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.

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

**Required:** Yes

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

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

**Required:** Yes

Location of the certificate (PEM-encoded) used by the FG Move server to contact with the Move site controllers. Client certificates.

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

**Required:** Yes

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

---

**Sections** `[Move-<service>-<serviceID>]` such as `[Move-eucalyptus-indiaeuca]` or `[Move-Hpc-indiahpc]`

This sections are to define how to contact the remove site controllers (`RainMoveSiteServer.py` server)

**Option `address`** Type: String

**Required:** Yes

Address of the server where FG Move site server will be running.

**Option `port`** Type: Integer

**Required:** Yes

Port where the FG Move site server will be listening.

---



### Section [RainMoveSiteServer]

This section is used to configure the FG Move Site Server, which is the FG Move controller place in the management machines.

```
port = 56796 proc_max = 5 refresh = 20 log = movesiteserver.log log_level = debug max_wait = 1000
ec2varfile = ~/eucarc ca_cert=/etc/futuregrid/imdservers/cacert.pem certfile=/etc/futuregrid/imdservers/imdsert.pem
keyfile=/etc/futuregrid/imdservers/privkey.pem
```

**Option port**    **Type:** Integer

**Required:** Yes

Port where the FG Move 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 ec2varfile**    **Type:** EC2-file

**Required:** (Only for Cloud infrastructures)

Location of the EC2 environment variable file, which typically is `eucarc` or `novarc`.

**Option hpcproperties**    **Type:** String

**Required:** (Only for HPC infrastructures)

Properties of the machines that are added to the HPC infrastructure. It can be several properties separated by commas.

**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 Move 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 `[RainMove]`**

This section is used to configure FG Move.

**Option `port`** Type: Integer

**Required:** Yes

Port where the FG Move server will be listening.

**Option `serveraddr`** Type: String

**Required:** Yes

Address of the machine where the FG Move 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 Move client.

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

**Required:** Yes

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

---

### 3.2.5 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/`

### 3.2.6 Building the FG Move 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.

## 3.3 Others

### 3.3.1 Changelog and Release Notes

#### 0.1

Summary of the main functionality provided for this version:

- **Security**
  - Authentication is performed using the FutureGrid LDAP server
- **Move**
  - Manage Clusters, Nodes and Services via command line interface
  - Move resources from/to HPC, Eucalytpus and OpenStack

# DOWNLOAD

The latest version of the FutureGrid Move 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\\_move](http://pypi.python.org/pypi/futuregrid_move)

## 4.1 Sample files

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

## 4.2 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 Move, please use our help form at <https://portal.futuregrid.org/help>





---

# KNOWN PROBLEMS

Depending on the configuration of your machines, you may have problems installing some python modules. In particular, we have noticed 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)
```

- Load FutureGrid module on India.

FutureGrid module loads the appropriated python module. Therefore, if you have previously loaded a python module, you need to unload it. Then load futuregrid module again.

The error looks like:

```
[jdiaz@i136 ~]$ module load futuregrid
futuregrid version 1.1 loaded
euca2ools version 2.0.2 loaded
python_w-cmd2/2.7(21):ERROR:150: Module 'python_w-cmd2/2.7' conflicts with the currently loaded
python_w-cmd2/2.7(21):ERROR:102: Tcl command execution failed: conflict python

moab version 5.4.0 loaded
torque/2.5.5 version 2.5.5 loaded
[jdiaz@i136 ~]$
```

The solution is:

```
[jdiaz@i136 ~]$ module unload python/2.7
[jdiaz@i136 ~]$ module load futuregrid
```