

## Resource Configuration

The configuration file `resources.conf` is used to describe computing resources. When you start DRM4G, `resources.conf` file is copied under `~/.drm4g/etc` directory by default if it does not exist or under whatever directory specified with **DRM4G\_DIR**. The file can be edited directly or by executing the `drm4g resource edit` command.

### Configuration format

The configuration resource file consists of sections, each led by a `[section]` header, followed by `key = value` entries. Lines beginning with `#` are ignored. Permitted sections are `[DEFAULT]` and `[resource_name]`.

### DEFAULT section

The DEFAULT section provides default values for all other resource sections.

### Resource section

Each resource section has to begin with the line `[resource_name]` followed by `key = value` entries.

The name of a resource cannot include this character `:`. The DRM4G won't be able to send jobs to a resource with that character in its name.

### Configuration keys common to all resources:

- `enable`: true or false in order to enable or disable a resource.
- `communicator` or authentication type :
  - `local`: The resource will be accessed directly.
  - `ssh`: The resource will be accessed through ssh's protocol via Paramiko's API.
  - `op_ssh`: The resource will be accessed through OpenSSH's CLI.
- `username`: Name of the user that will be used to log on to the front-end.
- `frontend`: Hostname or ip address of either the cluster or grid user interface you'll be connected to. The syntax is `"host:port"` and by default the port used is 22.
- `private_key`: Path to the identity file needed to log on to the front-end.
- `public key`: Path to the public identity file needed to log on to the front-end.
  - **OPTIONAL**: by default the `private_key`'s value will be taken, to which `.pub` will be added)
- `scratch`: Directory used to store temporary files for jobs during their execution, by default, it is `$HOME/.drm4g/jobs`
- `lrms` or Local Resource Management System :
  - `pbs`: TORQUE/PBS cluster.
  - `sge`: Grid Engine cluster.
  - `loadleveler`: LoadLeveler cluster.
  - `lsf`: LSF cluster.
  - `fork`: SHELL.
  - `cream`: CREAM Compute Elements (CE).
  - `slurm`: SLURM cluster.
  - `slurm_res`: [?RES\(Red Española de Supercomputación\)](#) resources.
  - `fedcloud`: it will indicate the DRM4G that this resource will be used to create VMs (Virtual Machines).

Note that for `communicator` you have two options when it comes to accessing a resource through the ssh protocol. If you don't know which one you prefer use `ssh`.

### Keys for non-grid resources such as HPC resources:

- `queue`: Queue available on the resource. If there are several queues, you have to use a `,` as follows `"queue = short,medium,long"`.
- `max_jobs_in_queue`: Max number of jobs in the queue.
- `max_jobs_running`: Max number of running jobs in the queue.
- `parallel_env`: It defines the parallel environments available for Grid Engine cluster.
- `project`: It specifies the project variable and is for TORQUE/PBS, Grid Engine and LSF clusters.

**Keys for grid resources:**

- `vo`: Virtual Organization (VO) name.
- `host_filter`: A host list for the VO. Each host is separated by a ",". Here is an example: "`host_filter = prod-ce-01.pd.infn.it, creamce2.gina.sara.nl`".
- `bdi`: It indicates the BDI host to be used. The syntax is "`bdi:port`". If you do not specify this variable, the `LCG_GFAL_INFOSYS` environment variable defined on the grid user interface will be used by default.
- `myproxy_server`: Server to store grid credentials. If you do not specify this variable, the `MYPROXY_SERVER` environment variable defined on the grid user interface will be used by default.

**Keys for FedCloud resources:**

**FedCloud** resources are a bit different than the rest. They are not considered to be `hosts` so they won't be listed when using the command `drm4g host` and therefore won't be used to execute jobs. Their only function is to connect to a machine with a cloud proxy certificate (**X.509 certificates**) capable of giving you access to cloud resources.

`vm_communicator`: or authentication type for the created VMs :

- `ssh`: The resource will be accessed through ssh's protocol via Paramiko's API.
- `op_ssh`: The resource will be accessed through OpenSSH's CLI.
- `vm_user`: Name of the user that will be used to log on to the creates VMs.
- `myproxy_server`: Server to store cloud credentials. If you do not specify this variable, the `MYPROXY_SERVER` environment variable defined on the grid user interface will be used by default.
- `nodes`: It indicates how many VMs you wish to create with the specified configuration
- `volume`: It's possible to create some extra storage and add it to the VM. With this you can specify how many extra GBs of storage you want.

The values of the next configuration keys can be customized at your discretion. A new cloud configuration file has been added to the DRM4G called "**cloudsetup.json**" for this reason. This resource keys reference the information saved in this cloud configuration file.

- `cloud`: Name that describes the site from which the image, that will be used to create the VM, will be acquired.
- `virtual_image`: It indicates which one of the system images available you will be using
- `flavour`: It indicates the hardware template for the VM

Where and how to get the correct values for your cloud configuration file as well as a more in depth explanation of some of these configuration keys can be found in the section [How to configure an EGI FedCloud VM](#).

A few extra things to take into consideration:

- If no `vm_user` is specified, `drm4g_admin` will be used by default.
- If no `vm_communicator` is specified, the one in `communicator` will be used, but if it's set to `local`, the DRM4G will set it to `ssh`.
- For the moment, the `lrms` for all created VMs will be `fork`.

The private key used to access the *VM* will be the same as the one used to access the machine that will create it.

- So even if you're going to use your local machine to create the VM, you'll have to specify a `private_key`.

**Examples**

By default, DRM4G is going to use the local machine as fork lrms:

```
[localmachine]
enable          = true
communicator    = local
frontend        = localhost
lrms            = fork
max_jobs_running = 1
```

TORQUE/PBS cluster, accessed through ssh protocol:

```
[meteo]
enable          = true
communicator    = ssh
```

```

username      = user
frontend      = mar.meteo.unican.es
private_key   = ~/.ssh/id_rsa
lrms          = pbs
queue        = short, medium, long
max_jobs_running = 2, 10, 20
max_jobs_in_queue = 6, 20, 40

```

SGE cluster, accessed through ssh protocol:

```

[blizzard]
enable      = true
communicator = op_ssh
username    = user
frontend    = blizzard.meteo.unican.es
private_key = ~/.ssh/id_rsa
parallel_env = mpi
lrms       = sge
queue      = long
max_jobs_running = 20
max_jobs_in_queue = 40

```

ESR virtual organization, accessed through a grid user interface:

```

[esrVO]
enable      = true
communicator = local
username    = user
frontend    = ui.meteo.unican.es
lrms       = cream
vo         = esr
bdii       = bdii.grid.sara.nl:2170
myproxy_server = px.grid.sara.nl

```

FedCloud virtual organization

```

[cesnet_metacloud]
enable      = true
communicator = ssh
username    = user
vm_communicator = op_ssh
vm_user     = drm4g_admin
frontend    = ui.meteo.unican.es
private_key = ~/.ssh/id_rsa
lrms       = fedcloud
cloud      = EGI FedCloud - CESNET-METACLOUD
myproxy_server = myproxyl.egee.cesnet.cz
flavour    = Small
virtual_image = Ubuntu-14.04
nodes      = 1
volume     = 0

```