

## Resource Configuration

The configuration file `resources.conf` is used to describe computing resources. When you start WRF4G, `resources.conf` file is copied under `~/wrf4g/etc` directory if it does not exist. The file can be edit directly or by executing `wrf4g 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. Allowing 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.

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 protocol.
- `username`: username to log on the front-end.
- `frontend`: The front-end either of a cluster or a grid user interface . The syntax is "host:port", by default the port used is 22.
- `private_key`: Private key identity file to log on the front-end.
- `scratch`: Shared directory used to store temporary files for jobs during their execution on the frontend, by default, it is `$HOME/.wrf4g/jobs`.
- `local_scratch`: Job's working directory on the worker nodes, by default, it is `$HOME/.wrf4g/jobs`.
- `lrms` or Local Resource Management System :
  - `pbs`: TORQUE/PBS cluster.
  - `sge`: Grid Engine cluster.
  - `slurm`: SLURM cluster.
  - `slurm_res`: [?RES\(Red Española de Supercomputación\)](#) resources.
  - `loadleveler`: LoadLeveler cluster.
  - `lsf`: LSF cluster.
  - `fork`: SHELL.
  - `cream`: CREAM Compute Elements (CE).

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 BDII 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.

### Examples

By default, WRF4G 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
```

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
```

## Example usage

### How to configure a TORQUE/PBS resource

In order to configure a TORQUE/PBS cluster accessed through ssh protocol, you should follow the next steps:

Configure the `meteo` resource. If you do not have a `private_key` file, you can generate one by executing [?ssh-keygen](#). This command will generate a public key (`~/.ssh/id_rsa.pub`) that will be necessary later on.

```
[user@mycomputer~]$ wrf4g resource edit
[meteo]
enable          = true
communicator    = ssh
username       = user
local_scratch  = $TMPDIR
frontend       = mar.meteo.unican.es
private_key    = ~/.ssh/id_rsa
lrms           = pbs
queue         = short
max_jobs_running = 2
max_jobs_in_queue = 6
```

2. List and check if resource has been created successfully :

```
[user@mycomputer~]$ wrf4g resource list
RESOURCE          STATE
```

```
meteo          enabled
```

Configure identity's resource copying the public key (~/.ssh/id\_rsa.pub) to authorized\_keys file on the remote frond-end, and adds the private key to the agent for the ssh authorization:

```
[user@mycomputer~]$ wrf4g id meteo init
--> Starting ssh-agent ...
--> Adding private key to ssh-agent ...
    Identity added: /home/user/.ssh/id_rsa (/home/user/.ssh/id_rsa)
    Lifetime set to 7 days
--> Copying public key on the remote frontend ...
```

That's it! Now, you can submit experiments to meteo.

## How to configure a EGI VO

For configuring a EGI VO such as ESR accessed through user grid interface, follow the below steps:

Configure the esrVO resource. If the grid user interface has defined LCG\_GFAL\_INFOSYS and MYPROXY\_SERVER variables, you do not have to indicate bdii and myproxy\_server keys in your configuration:

```
[user@mycomputer~]$ drm4g resource edit
[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
```

- List and check if resource has been created successfully :

```
[user@mycomputer~]$ drm4g resource list
RESOURCE        STATE
esrVO           enabled
```

List the CEs available on the esr VO:

```
[user@mycomputer~]$ drm4g host list
HID ARCH      JOBS(R/T) LRMS      HOST
0   x86_64      0/0  cream-pbs esrVO::cream.afroditi.hellasgrid.gr
1   x86_64      0/0  cream-pbs esrVO::cel.ipgp.fr
2   x86_64      0/0  cream-pbs esrVO::crl.ipp.acad.bg
3   x86_64      0/0  cream-pbs esrVO::sbgce2.in2p3.fr
4   x86_64      0/0  cream-pbs esrVO::ce0.bordeaux.inra.fr
5   x86_64      0/0  cream-pbs esrVO::cce.ihep.ac.cn
6   x86_64      0/0  cream-pbs esrVO::ce02.ngcc.acad.bg
7   x86_64      0/0  cream-pbs esrVO::ce01.macc.unican.es
8   x86_64      0/0  cream-pbs esrVO::cygnus.grid.rug.nl
9   x86_64      0/0  cream-pbs esrVO::t2ce06.physics.ox.ac.uk
10  x86_64      0/0  cream-lsf esrVO::cel.ts.infn.it
11  x86_64      0/0  cream-lsf esrVO::gridcel.pi.infn.it
12  x86_64      0/0  cream-lsf esrVO::gridce3.pi.infn.it
13  x86_64      0/0  cream-pbs esrVO::cream02.grid.uoi.gr
14  x86_64      0/0  cream-pbs esrVO::lapp-ce02.in2p3.fr
15  x86_64      0/0  cream-pbs esrVO::grid002.jet.efda.org
16  x86_64      0/0  cream-lsf esrVO::gridce4.pi.infn.it
```

17	x86_64	0/0	cream-lsf	esrVO::gridce0.pi.infn.it
18	x86_64	0/0	cream-lsf	esrVO::gridce2.pi.infn.it
19	x86_64	0/0	cream-pbs	esrVO::t2ce06.physics.ox.ac.uk
20	x86_64	0/0	cream-pbs	esrVO::grid0.fe.infn.it
21	x86_64	0/0	cream-pbs	esrVO::ce0.m3pec.u-bordeaux1.fr
22	x86_64	0/0	cream-pbs	esrVO::juk.nikhef.nl
23	x86_64	0/0	cream-pbs	esrVO::gridce.ilc.cnr.it
24	x86_64	0/0	cream-lsf	esrVO::cert-37.pd.infn.it
25	x86_64	0/0	cream-pbs	esrVO::cream-ce-2.ba.infn.it
26	x86_64	0/0	cream-sge	esrVO::ccccreamceli09.in2p3.fr
27	x86_64	0/0	cream-sge	esrVO::ccccreamceli10.in2p3.fr
28	x86_64	0/0	cream-pbs	esrVO::gazon.nikhef.nl
29	x86_64	0/0	cream-pbs	esrVO::klomp.nikhef.nl
30	x86_64	0/0	cream-pbs	esrVO::cream-ce-3.ba.infn.it
31	x86_64	0/0	cream-pbs	esrVO::cream-ce-4.ba.infn.it
32	x86_64	0/0	cream-pbs	esrVO::creamce.gina.sara.nl
33	x86_64	0/0	cream-lsf	esrVO::prod-ce-01.pd.infn.it
34	x86_64	0/0	cream-pbs	esrVO::creamce2.gina.sara.nl
35	x86_64	0/0	cream-pbs	esrVO::creamce3.gina.sara.nl
36	x86_64	0/0	cream-slur	esrVO::ce3.ui.savba.sk
37	x86_64	0/0	cream-pbs	esrVO::glite-cream.scai.fraunhofer.de
38	x86_64	0/0	cream-pbs	esrVO::cream-ce02.marie.hellasgrid.gr
39	x86_64	0/0	cream-pbs	esrVO::cream-ce01.marie.hellasgrid.gr
40	x86_64	0/0	cream-pbs	esrVO::fal-pygrid-44.lancs.ac.uk
41	x86_64	0/0	cream-pbs	esrVO::hepgrid6.ph.liv.ac.uk
42	x86_64	0/0	cream-pbs	esrVO::cream-ce01.ariagni.hellasgrid.gr
43	x86_64	0/0	cream-pbs	esrVO::snf-189278.vm.oceanos.grnet.gr
44	x86_64	0/0	cream-pbs	esrVO::snf-458754.vm.oceanos.grnet.gr
45	x86_64	0/0	cream-pbs	esrVO::hepgrid5.ph.liv.ac.uk
46	x86_64	0/0	cream-pbs	esrVO::cream01.kallisto.hellasgrid.gr
47	x86_64	0/0	cream-pbs	esrVO::hepgrid10.ph.liv.ac.uk
48	x86_64	0/0	cream-pbs	esrVO::hepgrid97.ph.liv.ac.uk

#### 4. Create an identity for 7 days:

```
[user@mycomputer~]$ drm4g id esrVO init
drm4g id esrVO init
Starting ssh-agent ...
WARNING: ssh-agent is already running
--> Add '/home/user/.ssh/id_rsa' into ssh-agent for 168 hours
Lifetime set to 7 days, 0:00:00
--> Copy '/home/user/.ssh/id_rsa' to ~/.ssh/authorized_keys file on 'mar.meteo.unican.es'
WARNING: It is assumed that the grid certificate has been already configured
--> Creating '~/drm4g/security' directory to store the proxy ...
Starting ssh-agent ...
WARNING: ssh-agent is already running
--> Create a local proxy credential ...
Insert your Grid password:
Your identity: /DC=es/DC=irisgrid/O=unican/CN=user
Creating proxy ..... Done
Proxy Verify OK
Your proxy is valid until: Thu Feb 26 21:37:19 2015
Your identity: /DC=es/DC=irisgrid/O=unican/CN=user
Creating proxy ..... Done
Proxy Verify OK
A proxy valid for 168 hours (7.0 days) for user /DC=es/DC=irisgrid/O=unican/CN=user now exists on px.grid.sara.nl.
```

#### 5. Check the timeleft of your identity:

```
[user@mycomputer~]$ drm4g id esrVO info
--> Grid credentials
subject  : /DC=es/DC=irisgrid/O=unican/CN=user/CN=proxy/CN=proxy
issuer   : /DC=es/DC=irisgrid/O=unican/CN=user/CN=proxy
identity : /DC=es/DC=irisgrid/O=unican/CN=user
type     : full legacy globus proxy
strength : 2048 bits
path     : /home/user/.drm4g/security/px.grid.sara.nl
timeleft : 167:57:52 (7.0 days)
```

That's it! Now, you can submit jobs to the `esr` VO. Keep in mind that you will have to renew your identity depending on the `proxy-lifetime` used.