

How to configure an EGI FedCloud VM

For configuring an **EGI Federated Cloud Virtual Machine**, follow the steps below:

First, you'll need to have access to cloud resources. To do that you'll have to follow the instructions given to [?access the federated services](#).

Then, once you've correctly installed the **DRM4G**, in your `$DRM4G_DIR/etc` directory you'll find a file called `cloudsetup.json`. In it you'll have to define the information needed to create your Virtual Machine (VM).

Configuration format

- It's a JSON file, so its syntax includes the following:
 - Data is represented in name/value pairs.
 - Curly braces hold objects and each name is followed by ':' (colon), the name/value pairs are separated by ',' (commas).

By default it comes with examples using sites that have been proven to work. The list will be updated with upcoming releases.

This is an example of how the "`cloudsetup.json`" should look like:

```
{
  "rocci" : {
    "vo" : "fedcloud.egi.eu",
    "cloud_providers" : {
      "EGI FedCloud - CESNET-METACLOUD" : {
        "endpoint" : "https://carach5.ics.muni.cz:11443",
        "apps" : {
          "Ubuntu-14.04" : "http://occi.carach5.ics.muni.cz/occi/infrastructure/os_tpl#uuid_egi_ubuntu_server_14_04",
        },
        "flavours" : {
          "Small" : "http://fedcloud.egi.eu/occi/compute/flavour/1.0#small",
          "Medium" : "http://fedcloud.egi.eu/occi/compute/flavour/1.0#medium"
        }
      }
    }
  }
}
```

Cloud configuration keys

`rocci`: Indicates the name of the *Virtual Organization* (VO) being used. In the future more may be added.

`vo`: it's used for authorization, to get the proxy certificate

- `fedcloud.egi.eu`: shouldn't be changed for FedCloud sites

`cloud_providers`: it will usually contain a list of sites with the VM configurations.

`<site_name>`: name that represents the site where your VM will be hosted

- `endpoint`: end of a communication channel, URL of their server

`apps`: list of system images

- `<system_image_name>`: OCCI ID that represents the system image you'll be using

`flavours`: list of template IDs

- `<hardware_template>`: template ID that defines hardware specifications, sizes for RAM, disk and number of CPU cores

How to link the resource configuration file and the cloud setup file

There are a couple of configuration keys in "`resources.conf`" that reference values found in this JSON.

From the resource configuration file "`resources.conf`" you have to indicate which "`cloud_providers`", "`apps`" and "`flavours`" you want to use from all of the ones you have defined in the cloud setup file.

- It is done with the "`cloud_provider`", "`virtual_image`" and "`flavour`" keys from the resource configuration file respectively.

You can modify the name (not the value) of the "`<site_name>`", "`<system_image_name>`" and "`<hardware_template>`", so be sure to write the same value on the resource configuration file as the name on the cloud setup file.

The value in "resources.conf" has to match with the name in "cloudsetup.json"

```
[CESNET_Metacloud]
cloud_provider = EGI FedCloud - CESNET-METACLOUD
flavour       = Small
virtual_image  = Ubuntu-14.04

"cloud_providers" : {
  "EGI FedCloud - CESNET-METACLOUD" : {
    "endpoint" : "https://carach5.ics.muni.cz:11443",
    "apps" : {
      "Ubuntu-14.04" : "http://occi.carach5.ics.muni.cz/occi/infrastructure/os_tpl#uuid_egi_ubuntu_server_14_04",
    },
    "flavours" : {
      "Small" : "http://fedcloud.egi.eu/occi/compute/flavour/1.0#small",
      "Medium" : "http://fedcloud.egi.eu/occi/compute/flavour/1.0#medium"
    }
  }
}
```

How to manage VMs

Customize your cloudsetup.json to match your requirements. You can search for the available system images in [?EGI's Applications Database Cloud Marketplace](#).

- Search for the system image you want.

- Select the one you prefer.

 <p>Sahara Apache Storm 0.9.1 (Ubuntu 14.04) (qco...</p> <p>Sahara Apache Storm 0.9.1 from the OpenStack Community App Catalog</p> <p>33 visits</p>	 <p>Sahara Apache Spark 1.0.0 (Ubuntu 14.04) (qco...</p> <p>Sahara Apache Spark 1.0.0 from the OpenStack Community App Catalog</p> <p>48 visits</p>
 <p>EGI Ubuntu 14.04</p> <p>EGI endorsed Ubuntu 14.04 LTS</p> <p>EGI Ubuntu 14.04</p> <p>Download</p> <p>16 sites member of ▼</p> <p>★ ★ ★ ★ ★ 1.0 1 vote</p>	 <p>Ubuntu 14.04.2 LTS (Docker)</p> <p>Ubuntu 14.04.2 LTS with Docker preinstalled</p> <p>Download</p> <p>165 visits</p>

- In the page that has opened up, select one of the sites that offer that same image.

Site: TR-FC1-ULAKBIM (TR) ▼

Site: IFCA-LCG2 (ES) ▼

Site: CESNET-METACLOUD (CZ) ▼

Site: CYFRONET-CLOUD (PL) ▼

Site: INFN-PADOVA-STACK (IT) ▼

- A list of flavours will appear. Choose one and click on "get IDs".

Site: CESNET-METACLOUD (CZ) ▲

Image : ver.2016.11.11 Ubuntu 14.04 / x86_64 / VirtualBox ▲

Memory	Logical/Physical CPUs	Connectivity In/Out	OS Family	
237568	40/40	yes/yes	linux	get IDs
65536	32/32	yes/yes	linux	get IDs
65536	16/16	yes/yes	linux	get IDs
32768	32/32	yes/yes	linux	get IDs
32768	16/16	yes/yes	linux	get IDs
32768	8/8	yes/yes	linux	get IDs
16384	4/4	yes/yes	linux	get IDs
8192	8/8	yes/yes	linux	get IDs
8192	2/2	yes/yes	linux	get IDs
4096	4/4	yes/yes	linux	get IDs
4096	1/1	yes/yes	linux	get IDs
2048	2/2	yes/yes	linux	get IDs
1024	1/1	yes/yes	linux	get IDs

View related IDs

After a pop-up window will appear with the information you'll have to write on the cloud setup file.

CESNET-MetaCloud IDs

Image: ver.2016.11.11 - Ubuntu 14.04 / x86_64 / VirtualBox

Memory: 2048	CPUs: 2/2	InOut: yes/yes	OS: linux
--------------	-----------	----------------	-----------

Site endpoint: <https://carach5.ics.muni.cz:11443>

Template ID: <http://fedcloud.egi.eu/occi/compute/flavour/1.0#medium>

OCCI ID: http://occi.carach5.ics.muni.cz/occi/infrastructure/os_tpl#uuid_egi_ubuntu_server_14_04_its_fedcloud_warg_131

close

- Site endpoint** corresponds to endpoint
- Template ID** corresponds to <hardware_template>
- OCCI ID** corresponds to <system_image_name>

```
{
  "rocci" : {
    "vo" : "fedcloud.egi.eu",
    "cloud_providers" : {
      "EGI FedCloud - CESNET-METACLOUD" : {
        "endpoint" : "https://carach5.ics.muni.cz:11443",
        "apps" : {
          "Ubuntu-14.04" : "http://occi.carach5.ics.muni.cz/occi/infrastructure/os_tpl#uuid_egi_ubuntu_server_14_04",
        },
        "flavours" : {
          "Medium" : "http://fedcloud.egi.eu/occi/compute/flavour/1.0#medium"
        }
      }
    }
  }
}
```

Configure a "rocci" resource in `resources.conf`.

- The way to inform the *DRM4G* that a defined resource is going to be used to create VMs, is to set its `lrms` key to `rocci`.

This way, the *DRM4G* won't consider this resource as a *host* and so **it won't send it jobs to execute**.

The VMs created by this resource are what will be considered to be its *hosts*. That entails that **the VMs will not be listed by the command** `drm4g resource list`

```
[user@mycomputer~]$ drm4g resource edit
[CESNET_Metacloud]
enable           = true
communicator     = ssh
vm_communicator = op_ssh
private_key      = ~/.ssh/id_rsa
username         = user
vm_user          = drm4g_admin
frontend         = ui.meteo.unican.es
lrms             = rocci
cloud_provider   = EGI FedCloud - CESNET-METACLOUD
myproxy_server  = myproxyl.egee.cesnet.cz
flavour          = Medium
virtual_image    = Ubuntu-14.04
instances        = 1
volume           = 10
max_jobs_running = 5
```

- Start the DRM4G and check if the resource has been created successfully :

```
[user@mycomputer~]$ drm4g start
Starting DRM4G ....
OK
[user@mycomputer~]$ drm4g resource list
RESOURCE          STATE
CESNET_Metacloud  enabled
```

- Create your remote proxy certificate

```
[user@mycomputer~]$ drm4g id CESNET_Metacloud init
--> Add '/home/user/.ssh/id_rsa' into ssh-agent for 168 hours
Lifetime set to 7 days, 0:00:00
--> Copying '/home/user/.ssh/id_rsa' to ~/.ssh/authorized_keys file on 'ui.meteo.unican.es'
The copy of the public key /home/user/.ssh/id_rsa.pub has been succesfull
WARNING: It is assumed that the grid certificate has been already configured
--> Creating '~/drm4g/security' directory to store the proxy ...
```

```

Connected (version 2.0, client OpenSSH_5.3)
Authentication (publickey) successful!
--> Create a local proxy credential ...
Insert your Grid password:
Your identity: <user_information>
Creating proxy ..... Done
Proxy Verify OK
Your proxy is valid until: Thu Dec  8 12:00:00 2016
Your identity: <user_information>
Creating proxy .....
Proxy Verify OK
A proxy valid for 168 hours (7.0 days) for user <user_information> now exists on myproxyl.egee.cesnet.cz.

Warning: your certificate and proxy will expire Thu Dec  8 12:00:00 2016
which is within the requested lifetime of the proxy

```

5. Create the VM (Virtual Machine) :

This command will create as many VMs as you have defined in "[resources.conf](#)" **each time you execute it**

```

[user@mycomputer~]$ drm4g resource create
'ssh-agent' is running but without any keys
Connected (version 2.0, client OpenSSH_5.3)
Authentication (publickey) successful!
Creating new resource
The proxy '~/drm4g/security/x509up.fedcloud.ege.eu' has probably expired
Renewing proxy certificate
The proxy certificate will be operational for 24 hours
  Resource 'https://carach5.ics.muni.cz:11443/compute/79896' has been successfully created
Waiting until resource is active
Getting resource's IP direction
  Public IP: XXX.XXX.XXX.XX

```

Verify that the VM has been created and is been listed as a *host* :

```

[user@mycomputer~]$ drm4g host
HID ARCH      JOBS(R/T) LRMS      HOSTNAME
1  x86_64      0/0 fork    CESNET_Metacloud_XXX.XXX.XXX.XX

```

7. To send it jobs, just submit some as you'd normally do:

```

[user@mycomputer~]$ drm4g job submit path/to/job.job --ntasks 10
ARRAY ID: 0

TASK JOB
0  0
1  1
2  2
3  3
4  4
5  5
6  6
7  7
8  8
9  9

```

8. And then to check that it is actually doing them:

```
[user@mycomputer~]$ drm4g job list
```

JID	DM	EM	START	END	EXEC	XFER	EXIT	NAME	HOST
0	done	----	12:07:43	12:08:14	0:00:12	0:00:07	0	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork
1	done	----	12:07:43	12:08:14	0:00:12	0:00:07	0	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork
2	done	----	12:07:43	12:08:15	0:00:21	0:00:07	0	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork
3	done	----	12:07:43	12:08:14	0:00:21	0:00:06	0	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork
4	done	----	12:07:43	12:08:15	0:00:21	0:00:07	0	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork
5	wrap	pend	12:07:43	--:--:--	0:00:04	0:00:01	--	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork
6	wrap	pend	12:07:43	--:--:--	0:00:05	0:00:00	--	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork
7	wrap	pend	12:07:43	--:--:--	0:00:04	0:00:01	--	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork
8	wrap	----	12:07:43	--:--:--	0:00:01	0:00:00	--	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork
9	wrap	----	12:07:43	--:--:--	0:00:00	0:00:01	--	date.job	CESNET_Metacloud_XXX.XXX.XXX.XX/fork

To list resources, you'd usually use the command `drm4g resource list`, but that doesn't show the created VMs. If you want to see all of them listed, including the created VMs, in case you want to manually connect to it for example, you can use the command option `"--all"`:

```
[user@mycomputer~]$ drm4g resource list --all
```

```
Resources:
```

```
  CESNET_Metacloud
    communicator:  ssh
    username:      user
    frontend:      ui.meteo.unican.es
    private key:   /home/user/.ssh/id_rsa
    lrms:          rocci
  CESNET_Metacloud_XXX.XXX.XXX.XX
    communicator:  op_ssh
    username:      drm4g_admin
    frontend:      XXX.XXX.XXX.XX
    private key:   /home/user/.ssh/id_rsa
    lrms:          fork
```

1. Finally, once you're done with your VM, destroy it.
 - This command will destroy all of the VMs you've created so far:

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

```
Deleting resource https://carach5.ics.muni.cz:11443/compute/79896
```

```
Connected (version 2.0, client OpenSSH_5.3)
```

```
Authentication (publickey) successful!
```

```
Resource 'https://carach5.ics.muni.cz:11443/compute/79896' has been successfully deleted
```

That's it! Now, you can submit jobs to **VMs** created using the **EGI Federated Cloud VO**.