



EUROPEAN MIDDLEWARE INITIATIVE

GENERIC INSTALLATION & CONFIGURATION FOR EMI 1

Document Version:	1.0.0
Date:	11.05.2011

Table of Contents

Generic Installation & Configuration for EMI 1	1
Installing the Operating System	2
Scientific Linux 5.....	2
Node synchronization, NTP installation and configuration.....	2
Cron and logrotate.....	3
Host Certificates	4
Installing the Middleware	5
Repositories.....	5
The middleware (EMI) repository.....	5
The Certification Authority repository.....	6
The EPEL repository.....	6
Installations.....	6
Configuring the Middleware	8
Using the YAIM configuration tool.....	8
Configuration information.....	8
The LSF batch system.....	9
The CREAM for LSF.....	10
The WN for LSF.....	10
Note on site-BDII for LSF.....	10
The Torque/PBS batch system.....	11
TORQUE Server.....	11
The WN for Torque/PBS.....	11
The UL.....	11

Generic Installation & Configuration for EMI 1

This document is addressed to Site Administrators responsible for middleware installation and configuration. It is a generic guide to manual installation and configuration for any supported node types.

The list of supported node types can be found in the EMI 1 web pages.

When installing a particular node type please also have a look at the specific release page of that node type to get specific installation information.

Installing the Operating System

All EMI 1 components are fully supported on the SL5/64 platform with EPEL as repository for external components.

Full platform support means the component is distributed from the EMI repository using certified source and binary packages according to the format specification of the platform.

Scientific Linux 5

The OS version of EMI 1 is Scientific Linux 5 (SL). For more information please check:
<http://www.scientificlinux.org>

All the information to install the operating system can be found: <https://www.scientificlinux.org/download>
Example of **sl.repo** file

```
[core]
name=name=SL 5 base
baseurl=http://linuxsoft.cern.ch/scientific/5x/$basearch/SL
    http://ftp.scientificlinux.org/linux/scientific/5x/$basearch/SL
    http://ftp1.scientificlinux.org/linux/scientific/5x/$basearch/SL
    http://ftp2.scientificlinux.org/linux/scientific/5x/$basearch/SL
protect=0
```

Node synchronization, NTP installation and configuration

A general requirement is that the nodes are synchronized. This requirement may be fulfilled in several ways. If your nodes run under AFS they are most likely already synchronized. Otherwise, you can use the NTP protocol with a time server.

Instructions and examples for a NTP client configuration are provided in this section. If you are not planning to use a time server on your machine you can just skip this section.

Use the latest ntp version available for your system. If you are using APT, an apt-get install ntp will do the work.

- Configure the file `/etc/ntp.conf` by adding the lines dealing with your time server configuration such as, for instance:

```
restrict <time_server_IP_address> mask 255.255.255.255 nomodify notrap noquery
server <time_server_name>
```

Additional time servers can be added for better performance results. For each server, the hostname and IP address are required. Then, for each time-server you are using, add a couple of lines similar to the ones shown above into the file `/etc/ntp.conf`.

- Edit the file `/etc/ntp/step-tickers` adding a list of your time server(s) hostname(s), as in the following example:

```
137.138.16.69
137.138.17.69
```

- If you are running a kernel firewall, you will have to allow inbound communication on the NTP port. If you are using iptables, you can add the following to `/etc/sysconfig/iptables`

```
-A INPUT -s NTP-serverIP-1 -p udp --dport 123 -j ACCEPT
-A INPUT -s NTP-serverIP-2 -p udp --dport 123 -j ACCEPT
```

Remember that, in the provided examples, rules are parsed in order, so ensure that there are no matching REJECT lines preceding those that you add. You can then reload the firewall

```
# /etc/init.d/iptables restart
```

- Activate the ntpd service with the following commands:

```
# ntpdate <your ntp server name>
# service ntpd start
# chkconfig ntpd on
```

- You can check ntpd's status by running the following command

```
# ntpq -p
```

Cron and logrotate

Many middleware components rely on the presence of cron (including support for /etc/cron.* directories) and logrotate. You should make sure these utils are available on your system.

Host Certificates

All nodes except UI, WN and BDII require the host certificate/key files to be installed. Contact your Certification Authority (CA) to understand how to obtain a host certificate if you do not have one already.

Once you have obtained a valid certificate:

- *hostcert.pem* - containing the machine public key
- *hostkey.pem* - containing the machine private key

make sure to place the two files in the target node into the */etc/grid-security* directory and check the access right for *hostkey.pem* is only readable by root and that the public key, *hostcert.pem*, is readable by everybody.

Installing the Middleware

For SL5 the yum package manager is considered the to be the default installation tool.

Repositories

For a successful installation, you will need to configure your package manager to reference a number of repositories (in addition to your OS);

The middleware (EMI) repository

All EMI components are distributed from a single repository having the following structure:

- EMI-production (stable), **EMI/{1,2,3}**:
 - ◆ stable and signed, well tested software components, recommended to be installed on production-sites;
- EMI-testing, **testing/{1,2,3}**:
 - ◆ packages that will become part of the next stable distribution; passed the certification and validation phase and are available for technical-previews
- `platform/{arch-s,tgz,SRPMS}/{base/updates/third-party}`

The packages are signed with the EMI gpg key, that can be downloaded from here The fingerprint of the key is:

```
pub 1024D/DF9E12EF 2011-05-04
    Key fingerprint = AC82 01B1 DD50 6F4D 649E DFFC 27B3 331E DF9E 12EF
uid                               Doina Cristina Aiftimiei (EMI Release Manager) <aiftim@pd.infn.it>
sub 2048g/C1E57858 2011-05-04
```

To use yum:

- install the emi-release package. It will install also the EMI public key - or -
- wget the yum repository files from the following web address and copy it in /etc/yum.repos.d:
 - ◆ emi1-base.repo

```
[EMI-1-base]
name=EMI 1 base Repository
baseurl=http://emisoft.web.cern.ch/emisoft/dist/EMI/1/s15/$basearch/base
priority=45
protect=1
enabled=1
```

- ◆ emi1-updates.repo

```
[EMI-1-updates]
name=EMI 1 updates
baseurl=http://emisoft.web.cern.ch/emisoft/dist/EMI/1/s15/$basearch/updates
protect=1
enabled=1
# To use priorities you must have yum-priorities installed
priority=45
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-emi
```

- ◆ emi1-third-party.repo

```
[EMI-1-third-party]
```

```

name=EMI 1 third-party
baseurl=http://emisoft.web.cern.ch/emisoft/dist/EMI/1/s15/$basearch/third-party
protect=1
enabled=1
# To use priorities you must have yum-priorities installed
priority=45

```

The Certification Authority repository

All the details on how to install the CAs can be found in EGI IGTF release pages. Example of **ca.repo** file:

```

[EGI-trustanchors]
name=EGI-trustanchors
baseurl=http://repository.egi.eu/sw/production/cas/1/current/
gpgkey=http://repository.egi.eu/sw/production/cas/1/GPG-KEY-EUGridPMA-RPM-3
gpgcheck=1
enabled=1

```

The EPEL repository

If not present by default on your SL5/x86_64 nodes, you should enable the EPEL repository

EPEL has an 'epel-release' package that includes gpg keys for package signing and repository information. Installing this package, epel-release-5-4.noarch.rpm, should allow you to use normal tools such as yum to install packages and their dependencies. By default the stable EPEL repo is enabled. Example of **epel.repo** file:

```

[extras]
name=epel
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=epel-5&arch=$basearch
protect=0

```

Installations

You need to have enabled only the above repositories (Operating System, EPEL, Certification Authority, EMI).

Example of a general installation of a node:

```

# yum update
# yum install lcg-CA
# yum install <meta-package name>

```

The table below lists the available EMI meta-packages:

Node Type	meta-package name	Comments
AMGA_postgresql	emi-amga-postgresql	Details - Section 2 "Installation"
APEL	emi-apel	Details - Section3 - "Installation Instructions"
ARC-CE	nordugrid-arc-compute-element	
ARC Clients	nordugrid-arc-client-tools	arc* tools ng* tools
ARC InfoSys	nordugrid-arc-information-index	
ARGUS	emi-argus	
BDII_site	emi-bdii-site	
BDII_top	emi-bdii-totp	
CLUSTER	emi-cluster	

CREAM	emi-cream-ce	Installation of CREAM
CREAM LSF module	emi-lsf-utils	
CREAM TORQUE module	emi-torque-utils	
DPM mysql	emi-se-dpm-mysql	specific installation
DPM disk	emi-se-dpm-disk	specific installation
GLEEXEC_wn	emi-glexec-wn	The GLEEXEC_wn should always be installed together with a WN. more details in the Section "Installation"
LB	emi-lb	more info
LFC mysql	emi-lfc-mysql	specific installation
LFC oracle	emi-lfc-oracle	specific installation
MPI_utils	glite-mpi	MPI implementation instalaltion specific MPI-start instalaltion
STORM_backend	emi-storm-backend-mp	
STORM_frontend	emi-storm-frontend-mp	
STORM_checksum	emi-storm-checksum-mp	
STORM_gridhttps	emi-storm-gridhttps-mp	
STORM_globus_gridftp	emi-storm-globus-gridftp-mp	
STORM_srm_client	emi-storm-srm-client-mp	
TORQUE WN config	emi-torque-client	
TORQUE server config	emi-torque-server	
UNICORE/X		
UNICORE-UCC		
UNICORE Gateway		
UNICORE-HILA		
UNICORE Registry		
UNICORE TSI		
UNICORE XUADB		
UNICORE UVOS		
VOMS_mysql	emi-voms-mysql	
VOMS_oracle	emi-voms-oracle	
WMS	emi-wms	

Configuring the Middleware

Using the YAIM configuration tool

Some of EMI services can be configured using the YAIM tool. For a detailed description on how to configure the middleware with YAIM, please check the YAIM guide.

The necessary YAIM modules needed to configure a certain node type are automatically installed with the middleware. However, if you want to install YAIM rpms separately, you can install them by running `yum install glite-yaim-<node-type>`. This will automatically install the YAIM module you are interested in together with `yaim core`, which contains the core functions and utilities used by all the YAIM modules.

Configuration information

The table below lists the configuration instructions for some of EMI services:

Node Type/Service	Comments
AMGA_postgresql	yaim configuration target "AMGA_postgresql" use yaim
APEL	yaim configuration target "APEL" use yaim
ARC-CE	Sections 9-11
ARC Clients	arc* tools ARC Client Configuration Section "Configuration"
ARC InfoSys	Details
ARGUS	yaim config target "ARGUS_server" use yaim ARGUS Deployment
BDII_site	yaim config target "BDII_site" use yaim
BDII_top	yaim config target "BDII_top" use yaim
CLUSTER	CLUSTER config
CREAM	yaim config target "creamCE" CREAM Configuration
CREAM LSF module	yaim config target "LSF_utils" use yaim
DPM mysql	yaim config target "SE_dpm_mysql" use yaim specific HEAD_node configuration
DPM disk	yaim config target "SE_dpm_disk" use yaim specific DISK_node configuration
GLEEXEC_wn	yaim config target "GLEEXEC_wn" use yaim The GLEEXEC_wn should always be installed together with a WN.
LB	yaim config target "LB" use yaim more info

LFC mysql	yaim config target "LFC_mysql" use yaim specific configuration
LFC oracle	yaim config target "LFC_oracle" use yaim specific configuration
MPI_utils	yaim config target 'MPI_utils" use yaim Config batch system details on MPI-start configuration
STORM_backend	yaim config target 'SE_storm_backend" use yaim
STORM_frontend	yaim config target 'SE_storm_frontend" use yaim
STORM_checksum	yaim config target 'SE_storm_checksum" use yaim
STORM_gridhttps	yaim config target 'SE_storm_gridhttps" use yaim
STORM_globus_gridftp	yaim config target 'SE_storm_globus_gridftp" use yaim
STORM_srm_client	
TORQUE WN config	yaim config target 'TORQUE_client" use yaim
TORQUE server config	yaim config target "TORQUE_server" use yaim
CREAM TORQUE module	yaim config target "TORQUE_utils" use yaim
UNICORE/X	
UNICORE-UCC	
UNICORE Gateway	
UNICORE-HILA	
UNICORE Registry	
UNICORE TSI	
UNICORE XUADB	
UNICORE UVOS	
VOMS_mysql	yaim config target 'VOMS_mysql" use yaim more information
VOMS_oracle	yaim config target 'VOMS_oracle" use yaim more information
WMS	yaim config target 'WMS" use yaim more details on WMS config file

The LSF batch system

You have to make sure that the necessary packages for submitting jobs to your LSF batch system are installed on your CE. By default, the packages come as tar balls. At CERN they are converted into rpms so that they can be automatically rolled out and installed in a clean way (in this case using Quattor).

Since LSF is a commercial software it is not distributed together with the gLite middleware. Visit the Platform's LSF home page for further information. You'll also need to buy an appropriate number of license keys before you can use the product.

The documentation for LSF is available on Platform Manuals web page. You have to register in order to be able to access it.

The CREAM for LSF

- follow the CREAM Configuration Guide

The WN for LSF

Apart from the LSF specific configurations settings there is nothing special to do on the worker nodes. \After installing:

```
# yum install emi-wn
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n WN
```

just use the plain WN configuration target.

```
/opt/glite/yaim/bin/yaim -c -s site-info.def -n WN
```

Note on site-BDII for LSF

When you configure your site-BDII you have to populate the [vomap] section of the `/etc/lcg-info-dynamic-scheduler.conf` file yourself. This is because LSF's internal group mapping is hard to figure out from yaim, and to be on the safe side the site admin has to crosscheck. Yaim configures the lcg-info-dynamic-scheduler in order to use the LSF info provider plugin which comes with meaningful default values. If you would like to change it edit the `/etc/glite-info-dynamic-lsf.conf` file. After YAIM configuration you have to list the LSF group - VOMS FQAN - mappings in the [vomap] section of the `/etc/lcg-info-dynamic-scheduler.conf` file.

As an example you see here an extract from CERN's config file:

```
vomap :
  grid_ATLAS:atlas
  grid_ATLASSGM:/atlas/Role=lcgadmin
  grid_ATLASPRD:/atlas/Role=production
  grid_ALICE:alice
  grid_ALICESGM:/alice/Role=lcgadmin
  grid_ALICEPRD:/alice/Role=production
  grid_CMS:cms
  grid_CMSSGM:/cms/Role=lcgadmin
  grid_CMSPRD:/cms/Role=production
  grid_LHCB:lhcb
  grid_LHCBSGM:/lhcb/Role=lcgadmin
  grid_LHCBPRD:/lhcb/Role=production
  grid_GEAR:gear
  grid_GEARSGM:/gear/Role=lcgadmin
  grid_GEANT4:geant4
  grid_GEANT4SGM:/geant4/Role=lcgadmin
  grid_UNOSAT:unosat
  grid_UNOSAT:/unosat/Role=lcgadmin
  grid_SIXT:sixt
  grid_SIXTSGM:/sixt/Role=lcgadmin
  grid_EELA:eela
  grid_EELASGM:/eela/Role=lcgadmin
  grid_DTEAM:dteam
```

```
grid_DTEAMSGM:/dteam/Role=lcgadmin
grid_DTEAMPRD:/dteam/Role=production
grid_OPS:ops
grid_OPSSGM:/ops/Role=lcgadmin
module_search_path : ../lrms:../ett
```

The Torque/PBS batch system

TORQUE Server

- if you want to have a dedicated node for the TORQUE server:

```
# yum install emi-torque-server emi-torque-utils
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n TORQUE_server -n TORQUE_utils
```

- if you want to install configure the TORQUE server on the same node as the CREAM Computing Element:

```
# yum install emi-cream-ce emi-torque-server emi-torque-utils
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n creamCE -n TORQUE_server -n TORQUE_utils
```

For more details see the CREAM System Administrator Guide

The WN for Torque/PBS

```
# yum install emi-wn emi-torque-client
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n WN -n TORQUE_client
```

The UI

```
# yum install emi-ui
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n UI
```

-- DoinaCristinaAiftimiei - 17-Apr-2011

This topic: EMI > GenericInstallationConfigurationEMI1

Topic revision: r15 - 11-May-2011 - 23:54:17 - DoinaCristinaAiftimiei

Copyright &© by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback