CREAM
Installation&Configuration

Sara Bertocco
INFN Padova
Documentation

🌟 Main reference guide:

https://twiki.cern.ch/twiki/bin/view/EMI/GenericInstallationConfigurationEMI3

🌟 Other documentation:

🌟 EMI
http://www.eu-emi.eu/products/-/asset_publisher/1gkD/content/cream-3#Documentation_AN3

🌟 EGI-UMD
http://repository.egi.eu/2013/02/18/cream-1-14-2/
Prerequisites: OS

Required: a standard
- 64 bit SL(C)5
- 64 bit SL(C)6

Linux distribution properly installed.

Check OS version installed with
```
cat /etc/redhat-release
```
 Repositories: No DAG, Yes EPEL

★ The DAG repository must be removed or deactivated:
   rm /etc/yum.repos.d/dag.repo or
   mv /etc/yum.repos.d/dag.repo /etc/yum.repos.d/dag.repo.remove
   or enabled=0 in /etc/yum.repos.d/dag.repo

★ The EPEL repository must be installed
   wget \
   http://fedora-mirror01.rbc.ru/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm \
   --no-check-certificate
   yum install epel-release-6-8.noarch.rpm --nogpgcheck -y
   (*) Use --nogpgcheck to avoid to import the epel gpg key
Repositories: EMI 3 distribution

EMI repositories can be installed manually (configuring yum .repo files and giving EMI repositories precedence over EPEL. Check guide) through emi-release package (suggested):

```

yum install ./emi-release-3.0.0-2.el6.noarch.rpm
```
Repositories: Certification Authorities

Complete information on Certification Authorities distribution:
https://wiki.egi.eu/wiki/EGI_IGTF_Release

We need:

```bash
wget http://repository.egi.eu/sw/production/cas/1/current/repo-files/EGI-trustanchors.repo -O /etc/yum.repos.d/EGI-trustanchors.repo
```
Installation: CA certificates

The CA certificates can be installed issuing:

```
yum install ca-policy-egi-core --nogpgcheck -y
```

(*) To install gpg key (so not needed --nogpgcheck):

```
http://repository.egi.eu/sw/production/cas/1/GPG-KEY-EUGridPMA-RPM-3
```
Installation: CREAM CE

The CREAM CE can be installed issuing:

```
yum install emi-cream-ce
```
Exercise 1: EMI3 CREAM CE Installation

Install the CREAM CE

Relevant files:
http://wiki.scc.kit.edu/gridkaschool/index.php/Exercise_1:_EMI3_CREAM_CE_Installation
Choose the Batch System

Supported batch systems:

🌟 LSF
  must be installed and configured by hand

🌟 PBS Torque
  yaim configuration provided

🌟 (S)GE
  yaim configuration provided

🌟 Condor
  supported in the BLAH component (the component that implements the interface with the batch system), but not in information providers and in yaim configuration modules

🌟 SLURM supported in the BLAH component, supported in information providers (not much stable), partially supported in yaim (check EMI products documentation)
Install the Batch System

We choose to install PBS Torque with torque master co-hosted with CREAM:

```bash
yum install emi-torque-server
yum install emi-torque-utils
```

(*) If you are running Torque, and your CREAM CE node is NOT the torque master, on the CE you install only the emi-torque-utils metapackage:

```bash
yum install emi-torque-utils
```
Enable munge (1)

MUNGE is an authentication service for creating and validating credentials. It is designed to be highly scalable for use in an HPC cluster environment.

* Check that munge is installed:
  
rpm -qa |grep munge
munge-libs-x.y.z
munge-x.y.z
Enable munge (2)

Enable munge on your torque cluster:

★ Install the munge package (if it is not installed) on your pbs_server, submission hosts and all worker node hosts in your cluster.
★ On one host generate a key with /usr/sbin/create-munge-key
★ Copy the key, /etc/munge/munge.key to your pbs_server, submission hosts and all worker node hosts on your cluster.
★ Pay attention the ownership of that file must be:
  -r-------- 1 munge munge 1024 Jan 03 09:57 munge.key
  if needed:
  chown munge:munge /etc/munge/munge.key
  chmod 400 /etc/munge/munge.key
★ Start the munge daemon on these nodes
  service munge start
  chkconfig munge on
Exercise 2:
Batch System Installation

Install the Batch System:
yum install emi-torque-server
yum install emi-torque-utils
/usr/sbin/create-munge-key
chown munge:munge /etc/munge/munge.key
chmod 400 /etc/munge/munge.key
service munge start
chkconfig munge on
scp /etc/munge/munge.key <in WNs>

Relevant files:
http://wiki.scc.kit.edu/gridkaschool/index.php/
Exercise_2:_Torque_Batch_System_Installation
CREAM configuration

🌟 Configuration tool: YAIM

The YAIM modules needed to configure each middleware component are automatically installed with the middleware.

🌟 Relevant configuration files are:

🌟 users.conf
🌟 groups.conf
🌟 wn-list.conf
🌟 site-info.def
🌟 vo.d/
🌟 services/glite-creamce
Configuration: files customization

An example of file to customize is in 
/opt/glite/yaim/examples

★ siteinfo/site-info.def
★ siteinfo/services/glite-creamce
★ vo.d/*
★ users.conf (described in users.conf.README)
★ groups.conf (described in groups.conf.README)
★ wn-list.conf (described in wn-list.conf.README)

IMPORTANT: The configuration files coming with the YAIM rpm are just examples! Please review them and edit your own in a safe location!

/opt/glite/yaim/* is overwritten on every yaim update
users.conf is described in users.conf.README
This file defines the pool account user mapping, i.e. the UNIX users to be created on the service nodes that need them (mainly CE and WNs).

File format (fields must not have any white space):
UID:LOGIN:GID1[,GID2,...]:GROUP1[,GROUP2,...]:VO:FLAG:

UID    = user ID
LOGIN  = login name
GID1   = primary group ID
GID2   = secondary group ID
GROUP1 = primary group
GROUP2 = secondary group
VO     = virtual organization
FLAG   = string to identify special users

Example:
45003:dteam003:45000:dteam:dteam::
Configuration: groups.conf customization

groups.conf is described in groups.conf.README
This file defines the VOMS group mapping, i.e. the user categories that must be accepted by the grid services provided by a site. It indicates for each category to which kind of local accounts the user should be mapped, where applicable.

File format (fields must not have any white space):
"VOMS_FQAN":GROUP:GID:FLAG:[VO]

VOMS_FQAN = VOMS proxy fully qualified attribute name
GROUP = UNIX group
GID = UNIX GID
FLAG = string to identify special users
VO = virtual organization (optional. It allows the VO to be specified explicitly, otherwise it will be derived from the VOMS FQAN)

Examples:
"/my-VO/ROLE=lcgadmin":::sgm:
"/my-VO/foobar/ROLE=admin":::foobar_admin:
"/my-VO/foobar/*":::foobar_group:
"/my-VO/*":::
Configuration: wn-list.conf customization

wn-list.conf is described in wn-list.conf.README
This file defines the list of WN hostnames (FQDN) in the site.
Hostnames have to be Fully Qualified Domain Names (FQDN)

Example:
gks-026.scc.kit.edu
Configure users, groups and worker node list

Create your own <yaim_site_info_dir>
Copy example files from examples to <yaim_site_info_dir>
Customize users.conf adding testers.eu-emi.eu users
Customize groups.conf adding testers.eu-emi.eu groups
Fill your wn-list.conf

Relevant files:
http://wiki.scc.kit.edu/gridkaschool/index.php/
Exercise_3:_Configuration_of_users,_groups_and_worker_node_list
Configuration: site files customization

site-info.def is the main configuration file needed to execute the yaim command. It contains the list of variables common to multiple node types in form of key-value pairs. Some comments and information to the variables are online in the example file https://twiki.cern.ch/twiki/bin/view/LCG/Site-\info_configuration_variables#site_info_def

vo.d directory was created to make the configuration of the DNS-like VOs easier. It contains a file name per VO whose name has to be the lower-cased version of the VO name. The matching file should contain the definitions for that VO. In case the VO is also defined in site-info.def, the vo.d file will overwrite the variables which are defined there.

services/glite-creamce Contains cream-ce node type specific variables, it contains a list of key-value pairs.
Configuration verification

🌟 Verify if all need variables are configured:

```
/opt/glite/yaim/bin/yaim -v \n  -s /root/siteinfo_dir/site-info.def
  -n creamCE
```

🌟 Verify the file syntax

```
source site-info.def
```
Exercise 4: Site-info customization

★ Edit and customize your site-info.def
★ Uncomment dteam VO parameters in your site-info.def
★ Add DECH VO configuration parameters
★ Edit and customize your services/glite-creamce
★ Verify the configuration

Documentation reference:
https://twiki.cern.ch/twiki/bin/view/LCG/
Site-info_configuration_variables#site_info_def

Relevant files:
http://wiki.scc.kit.edu/gridkاسchool/index.php/
Exercise_4:_Site-info_customization
Install the host certificate

The CREAM CE node requires the host certificate/key files to be installed. If you do not have one already, contact your national Certification Authority (CA) to understand how to obtain a host certificate.

Make sure to place the two files in the target node into the /etc/grid-security directory.

Set the proper mode and ownerships:

chown root.root /etc/grid-security/hostcert.pem
chown root.root /etc/grid-security/hostkey.pem
chmod 644 /etc/grid-security/hostcert.pem
chmod 400 /etc/grid-security/hostkey.pem
Exercise 5: CREAM+Torque configuration

Finally we are ready to configure:

```
/opt/glite/yaim/bin/yaim -c \
   -s /root/siteinfo_dir/site-info.def \
   -n creamCE \
   -n TORQUE_server -n TORQUE_utils
```

Relevant files:
http://wiki.scc.kit.edu/gridkaschool/index.php/Exercise_5:_EMI_3_CREAM_CE_%2B_Torque_Server_Configuration
Troubleshooting – Init scripts

★ /etc/init.d/tomcat6
To start/stop the CREAM service, it is just necessary to start/stop the CREAM container.
In case the new BLAH blparser is used, this will also start it (if not already running).

★ /etc/init.d/glite-ce-blah-parser
If necessary to explicitly start the new BLAH blparser

★ /etc/init.d/glite-ce-blparser
Used if the old BLAH blparser is used, before starting tomcat, start it on the BLPARSER_HOST

★ /etc/init.d/bdii
★ /etc/init.d/mysqld
★ /etc/init.d/glite-lb-locallogger
★ /etc/init.d/globus-gridftp
★ service gLite start/stop/restart
Relevant Configuration Files

- CREAM configuration file (by yaim-cream-ce) 
  /etc/glite-ce-cream/cream-config.xml
- BLAH configuration file (by yaim-cream-ce) /etc/blah.config
- BLparser configuration file (by yaim-cream-ce) /etc/blparser.conf
  (only for the old blparser).
- glexec configuration file (by yaim-cream-ce) /etc/glexec.conf
- LCAS configuration file for glexec (by yaim-cream-ce) 
  /etc/lcas/lcas-glexec.db
- LCMAPS configuration file for glexec (by yaim-cream-ce) 
  /etc/lcmaps/lcmaps-glexec.db
- LCAS configuration file for gridftp (by yaim-cream-ce) 
  /etc/lcas/lcas.db
  (only when Argus is not used).
- LCMAPS configuration file for gridftp (by yaim-core) 
  /etc/lcmaps/lcmaps.db
  (only when Argus is not used).
- ARGUS configuration file for gridftp ( 
  /etc/grid-security/gsi-pep-callout.conf)
  (only when the CREAM CE is configured to use ARGUS).
The relevant log files are:

- The tomcat log file
  /usr/share/tomcat5/logs/catalina.out
- The trustmanager log file
  /usr/share/tomcat5/logs/trustmanager.log
- The new BLAH blparser log files
  /var/log/cream/glite-ce-bnotifier.log
  /var/log/cream/glite-ce-bupdater.log
- The old BLAH blparser log files
  /var/log/cream/glite-xxxparser.log  xxx=batch sys
- The gridftp log files
  /var/log/globus-gridftp.log
  /var/log/gridftp-session.log
Troubleshooting – Log files (2)

The relevant log files are:

★ The CREAM log file
/var/log/cream/glite-ce-cream.log
/var/log/cream-es/glite-ce-cream-es.log

★ The verbosity can be increased modifying the files
/etc/glite-ce-cream/log4j.properties
/etc/glite-ce-cream-es/log4j.properties
changing:
   log4j.logger.org.glite=info, fileout
with:
   log4j.logger.org.glite=debug, fileout
You may also change the attributes log4j.appender.fileout.MaxFileSize
and log4j.appender.fileout.MaxBackupIndex to change the maximum
file size and the maximum number of log files to be kept.
Troubleshooting – Log files (3)

The relevant log files are:

The glexec log file
/var/log/messages (syslog)
or set by yaim
GLEEXEC_CREAM_LOG_DESTINATION
GLEEXEC_CREAM_LOG_FILE
GLEEXEC_CREAMLCASLCMAPS_LOG

The verbosity can be changed editing the glexec configuration file
/etc/glexec.conf
Important on automatic updates

An update of an RPM not followed by configuration can cause problems.
STRONG RECOMMENDATION:
NOT TO USE AUTOMATIC UPDATE PROCEDURE OF ANY KIND.
Questions ?