

YAIM configuration of the Information System

Note: please, check the complete proposal in this wiki page

Roadmap for releasing glite-CLUSTER.

Please, check the following **Technical Note from Stephen Burke, Flavia Donno and Maarten Litmaath to understand why there's a need for glite-CLUSTER.**

A roadmap has been now defined:

lcg CE

- Phase 1: Release of cluster configuration as an independent YAIM module to be deployed in the same lcg CE host -> short term: This is currently being tested by SA3 partners but will probably be never released since we are interested in the version that allows us to install also the cluster publisher in a different host than the lcg CE. It's anyway a good exercise to test the code.
- Phase 2: Release of cluster configuration as an independent YAIM module to be deployed in the a different host than the lcg CE host -> medium term: This implies a set of changes in yaim that need to be implemented. See Steve Traylen's table that describes all the needed steps to achieve this.

Affecting yaim:

- ◆ Task A: glite-wn-info command -> new yaim function *config_wn_info* to be included in the WN.
- ◆ Task B: Information service provider per cluster -> new yaim function *_config_info_service_rtpublish* already written by Steve.
- ◆ Task C: VO tag dir per cluster -> new yaim function *config_vo_tag_dir* to be included in the cluster.
- ◆ Improvements after Steve's tests of Phase 1 version of yaim cluster and yaim lcg ce.

cream CE

- Deployment of the cluster configuration in the cream CE -> medium term : This requires the cooperation with yaim cream ce maintainers. Some preliminary work has been done

Description of the problem

The information system is currently configured by YAIM with the help of a set of configuration variables. These variables are internally used by YAIM and are assigned to appropriate Glue attributes by the `config_gip_<node-type>` functions. `config_gip_<node-type>` functions create `/opt/glite/var/tmp/gip/ldif/glite-info-static-XXX.conf` files that together with a set of template files `/opt/glite/etc/GlueYYY.template` are used to build ldif files. The ldif files are used by the gip to publish the information.

Stephen Burke has recently implemented a new information provider (see this YAIM wiki) that gives to the gip the same information that used to be provided with the ldif files. When this information can be obtained directly from the existing configuration of the node type, the information provider is able to retrieve it by itself without the need of using the YAIM variables. YAIM implements a new function called `config_info_service_<node-type>` that creates the wrapper to call the information provider for each node type.

However, there are some cases where we think Stephen Burke's information provider can't be used.

In the following table, we present the current list of node types that are part of the information system. We describe the relationship with the existing Glue templates and whether we think the information provider can

be used or not:

YAIM config target	YAIM function	conf file	template file	SB
glite-BDII_top	config_gip_bdii_top	glite-info-service-bdii_top.conf	<i>GlueService.template</i>	yes
glite-BDII_site	config_gip_bdii_site	glite-info-service-bdii-site.conf	<i>GlueService.template</i>	yes
glite-BDII_site	config_gip_site	glite-info-site.conf	<i>GlueSite.template</i>	NO
lcg-CE	config_gip_ce	glite-info-static-cluster.conf, glite-info-static-ce.conf, glite-info-static-cesebind.conf	<i>GlueCluster.template</i> , <i>GlueCE.template</i> , <i>GlueCESEBind.template</i>	Do
creamCE	config_cream_gip	glite-info-static-cluster.conf, glite-info-static-ce.conf, glite-info-static-cesebind.conf	<i>GlueCluster.template</i> , <i>GlueCE.template</i> , <i>GlueCESEBind.template</i>	Do
glite-FTS2	config_gip_fts2	-	-	It u gli
glite-LB	config_gip_lb	service-org.glite.lb.Server.conf	<i>GlueService.template</i>	yes
glite-LFC_mysql	config_gip_lfc	glite-info-static-lfc.conf	-	It u lcg
glite-MON	config_gip_mon	glite-info-service-secondary-producer.conf, glite-info-service-primary-producer.conf, glite-info-service-consumer.conf, glite-info-service-on-demand-producer.conf, glite-info-service-browser.conf	<i>GlueService.template</i>	yes
glite-PX	config_gip_px	glite-info-static-px.conf	<i>GlueService.template</i>	yes
glite-SE_classic	config_gip_se_classic	glite-info-static-se.conf	<i>GlueSE.template</i>	yes
glite-SE_dpm_*	glite-se_dpm_mysql	glite-info-static-dse.conf, glite-info-static-se.conf	<i>GlueService.template</i> , <i>GlueSE.template</i>	It u gli
glite-SE_dcach	glite-se_dcach	lcg-info-static-dse.conf, lcg-info-static-se.conf	<i>GlueService.template</i> , <i>GlueSE.template</i>	yes
glite-WMS	glite-wms	service-org.glite.wms.WMProxy	<i>GlueService.template</i>	yes

We think that whenever we use the *GlueService.template*, SB's information providers are fine. This means we can get rid of the *ldif* files for most of the node types listed in the previous table.

However, there are some cases where it's not so clear if this is possible:

- lcg CE and cream CE
 - ◆ *GlueCE.template*: Yes with SB's (SB should confirm this)
 - ◆ *GlueCESEBind.template*: Yes with SB's (SB should confirm this)
 - ◆ *GlueCluster.template*: This will be in the future part of a new yaim module called yaim cluster. See Steve Traylen's presentation [☞](#) and a first attempt to implement his proposal in CVS [☞](#). We think it should be the sys admin who defines the information needed per cluster and subcluster. We think this is not possible to implement with the information provider.
- site BDII
 - ◆ *GlueSite.template*: It requires a description of the site that needs to be defined by the sys admin.

For these cases we need to improve the `config_gip_<node-type>` functions. As they are now, we do not allow to preserve manual modifications made by the sys admin.

We have thought of getting rid of the `site-info.def` variables and distribute instead a template of what YAIM creates now in `/opt/glite/var/tmp/gip/ldif/glite-info-static-XXX.conf`. But we don't see the advantage of this, since everytime a new variable is needed, we also need to distribute a new version of the template. We think we can keep on using configuration variables but improve the naming conventions. This presents some advantages.

We propose to define the name of the variables in the following way

<mandatory-prefix>_<variable-name>_<optional-suffix>. If I take the example of the new variables introduced by Steve Traylen in patch 1778[?](#), a variable shall be like `SITE_OtherInfo_GRID="WLCG|EGEE"`:

- the prefix (mandatory) is the type of Glue object related to the variable we are using (SITE, KEY, SUBCLUSTER, HOST, ...)
- the variable name is the real Glue name (*OtherInfo*, *PhysycalCPUs*, *TmpDir*, ...)
- the suffix (optional) is used when the value of a variable is a key/value pair, like in the case of

```
GlueSiteOtherInfo: GRID=WLCG|EGEE
```

Like this, we can create a generic function that will loop over the objects defined under a certain dn. For each of those objects, YAIM will try to find a variable (with the previous name convention) in the configuration directory. If it exists, it will assign that value. If it doesn't exist and there's a reasonable default value (we will have a list of default values in the corresponding `default/` YAIM file when necessary) YAIM will use that one. If there's no variable defined and no default, we will give an error.

Example for *GlueSite.template*

- Configuration variables

```
# Variables defined by the sys admin (eventually distributed under services/glite-bdii_site):
SITE_UniqueID=yaim-site.cern.ch
SITE_Name="YAIM site"
SITE_Description="This is the YAIM site"
SITE_UserSupportContact="yaim@cern.ch"
SITE_SysAdminContact="yaimadmin@cern.ch"
SITE_SecurityContact="yaimsafe@cern.ch"
SITE_Location=Geneva
SITE_Latitude=12
SITE_Longitude=12
SITE_Web=www.yaim.info
SITE_Sponsor=EGEE
SITE_OtherInfo_GRID="EGEE|WLCG"
SITE_OtherInfo_ROC=UK/I
```

- code to be included in `config_gip_site` (this is only some basic code to present this approach, it has to be completed and tested)

```
config_gip_site_check () {
    for i in ${!GlueSite*} do
        name=${i//GlueSite_/}
        requires $1 SITE_$NAME      # we need to improve this code to ONLY check mandatory variables
    done
}

config_gip_site () {
    outfile=$INSTALL_ROOT/glite/var/tmp/gip/glite-info-site.conf

    echo "dn:" >> $outfile
    for i in ${!SITE_*} do
        name=${i//SITE_/}
        echo "GlueSite$name: $i" >> $outfile      # special code would be needed for variables like SIT
    done
    echo "GlueForeignKey: GlueSiteUniqueID=SITE_SiteUniqueID" >> $outfile

    $INSTALL_ROOT/glite/sbin/glite-info-static-create -c $outfile -t \
    $INSTALL_ROOT/glite/etc/GlueSite.template > \
    $INSTALL_ROOT/glite/etc/gip/ldif/glite-info-site.ldif
```

}

The advantages of this approach are:

- Sys admin can include as many Glue variables as the need in the configuration as long as the use the naming convention properly. There's no longer need for a new bug everytime we want to configure a new Glue variable with YAIM. However, YAIM will try to provide example configuration files for all the existing Glue variables in the current Glue schema.
- Changing the gip configuration is as easy as changing the value of the configuration variables and re-running the relevant `config_gip<node-type>` function.
- We can make this approach and the current one live together for a while by implementing a "translator" which will rename old configuration variables to the new names. We can already re-write the `config_gip_<node-type>` functions to work only with the new names and use the translator for a transition period.

What we need now is:

- Stephen Burke should confirm our assumptions regarding his information provider.
- Check whether we can actually benefit from the naming convention in the scenarios where we can't use SB's information provider.
- Study carefully the transition period (only for 3.1. For 3.0 we won't modify anything).

-- MariaALANDESPRADILLO - 15 Apr 2008

This topic: EGEE > YAIMInfosys

Topic revision: r12 - 2011-02-08 - MariaALANDESPRADILLO



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

Ideas, requests, problems regarding TWiki? Ask a support question or Send feedback