Table of Contents

WARNING ! This page is now deprecated - please see GridMonitoringNcgYaim

NCG Overview

The following pages describe the use of the WLCG Grid Services Monitoring Working Group site service monitoring configurator: NCG.

NCG is a three phase configuration generator for automation of the production of monitoring configurations for Grid sites and their services. The "N" in NCG originally stood for "Nagios", since this was the original target monitoring system. NCG now has more general application through a modular, extensible design with Nagios being the default target.

The three phases of NCG are as follows -

1. Gather information about all hosts and Grid services associated with a named Site. This is referred to as *topology information* gathering and is primarily derived from the Information System(BDII), the SAM database and locally site-defined data held in text files.
2. Merge the topology information with data defining tests (*probe description database*) which are appropriate for gathering metrics of the state of each type of Grid service. After this merging a complete map of the site monitoring system is available.
3. Use the output map from the two phases to generate configuration files for a specific target monitoring tool (e.g. Nagios). Conceptually, this is the only phase which is dependent on the target tool.

The following sections cover manual installation of a Nagios Grid Service monitor. They cover simple installation from scratch so will have to be adapted by administrators with existing Nagios installations. Please send feedback to wlcg-monitoring-discuss@cernNOSPAMPLEASE.ch (register here) or file a Savannah bug.

Installation via Yaim modules is available here.

Probe type options

Throughout this guide we distinguish between three classes of probe: *local*, *native* and *remote*.

In this context **remote** means probes which are executed against your site services by some external agent. Two such external agents are the central SAM grid monitoring services and the network monitoring probes run by the ENOC. Both these central services test site services and publish the results through well defined interfaces. Configuring your site monitoring to use *remote* probes means your site monitoring fetches the results from the central service and displays or acts upon the state determined by the external agent. In the case of Nagios these *remote* results are displayed as *passive service checks*.

Local probes are tests which a site monitoring service schedules itself and result in some interaction between the monitoring service and the monitored grid service, generally through the execution of a command on a User Interface to check some functionality against and expected result or testing at a lower level of, for instance, a service listening on a specific port. In the *Nagios* sense, *local* probes are displayed as *active service checks*.

Native probes are native Nagios checks which are used to actively check monitored grid service. These probes are executed and displayed in the same way as local probes. These probes are part of standard Nagios installation and don't require any additional packages to be installed.

Currently we **recommend** site administrators new to NCG to start with the 'remote,native' configuration, and to try out the 'all' configuration on a fresh Nagios instance for testing purposes.

Using remote gLite UI via NRPE

Local probes require User Interface (UI) middleware to be deployed on Nagios server. If this is not an option, admins can use existing UI node on site for running local probes. In the following sections specific actions needed in case when such NRPE UI is used will be emphasized.

Nagios Grid Service monitor installation

Following configurations are possible:

- 'remote' - only the results from remote monitoring systems (SAM and ENOC Downcollector) tests are added to the configuration
- 'local' - only results of WLCG probes run by the local monitoring system are added to the configuration. This requires a security configuration described below.
- 'native' - only results of Nagios native checks run by the local monitoring system are added to the configuration. Examples are: check_tcp, check_ldap, check_ssh.
- mix of options described above can be listed (separated with comma)
- 'all' - all options described above are enabled.

RPM Installation Repositories

Important! Repository location has changed. WLCG monitoring repo (described here) is obsolete, please use new one instead.

Important! Directory hierarchy of new RPMs has changed. Files are stored in standard OS directories (/usr/...) and not in /opt/lcg. Please update your old ncg.conf files accordingly (e.g. TEMPLATES_DIR).

In order to install the egee-SA1 repository, create a file with the following contents in /etc/yum.repos.d/egee-SA1.repo:

```
[egee-SA1]
name=EGEE SA1 software
baseurl=http://www.sysadmin.hep.ac.uk/rpms/egee-SA1/sl4/$basearch/
enabled=1
gpgcheck=0
```

Prerequisites

In order to generate Nagios configuration for your site and run remote probes against central SAM service, you need to request access to SAM PI from your Nagios host. Details on enabling access are maintained by the SAM team here. In the request you should provide the machine address(es) and simply specify that you require access under the "WLCG Grid Services Monitoring Profile".

Install prerequisites -

- install Apache httpd (required by Nagios)
- install Nagios and Nagios plugins
  RPMs `nagios` and `nagios-plugins` from DAG RPM repository
- if you are configuring for 'Local tests' then install and configure gLite-UI node now (note the prerequisite for java SDK)

- the following perl modules may also be required depending on your base distribution:
  `Date::Parse Date::Format (RPM perl-TimeDate)`
  `LWP::UserAgent (RPM perl-libwww-perl)`
  `Crypt::SSLeay (RPM perl-Net-SSLeay)`
  `XML::Parser (RPM perl-XML-Parser)`
  `XML::Writer (RPM perl-XML-Writer)`
  `CGI (RPM perl-CGI)`
  `Config::General (RPM perl-Config-General)`
- local service probes require that the local firewall on the machine must open ports specified by GLOBUS_TCP_PORT_RANGE.

For running local probes the system relies on being run on a gLite User Interface (UI) node, and uses part of the environment set by default on this node type. Alternative option is to use NRPE UI.

If NRPE UI is used, on UI server install prerequisites:

- install NRPE service
  RPMs `nagios-nrpe` from DAG RPM repository
- local service probes require that the local firewall on the machine must open ports specified by GLOBUS_TCP_PORT_RANGE.

Installation

A minimal install consists of the WLCG Nagios configuration generator (`grid-monitoring-config-gen`) which is used to create the WLCG-specific configuration files and the Nagios implementation of the prototype configured either for the fetching of only the remote probe (e.g SAM) results or for both remote and locally-run service probes (`grid-monitoring-fm-nagios-{remote,local}`).

- Install WLCG Nagios Configuration Generator
  Package name : `grid-monitoring-config-gen`

- Install WLCG monitoring plugins for Nagios as required
  - For fetching of remote probe results
    Package name : `grid-monitoring-fm-nagios-remote`
  - For the execution of local service probes (optional)
    Package name : `grid-monitoring-fm-nagios-local`
    Package name : `grid-monitoring-probes-ch.cern`
    Package name : `grid-monitoring-probes-hr.srce`
    Package name : `nagios-proxy-refresh` (**only in new EGEE SA1 repo**)
    Package name : `msg-nagios-bridge` (**only in new EGEE SA1 repo**)

- If NRPE UI is used, install following packages on UI node:
  Package name : `grid-monitoring-fm-nagios-local`
  Package name : `grid-monitoring-probes-ch.cern`
  Package name : `grid-monitoring-probes-hr.srce`
  Package name : `nagios-proxy-refresh` (**only in new EGEE SA1 repo**)

Proxy Certificate Security Configuration

The following security configuration is **only** necessary if you are planning to configure for **local probes**.

New procedure

This procedure should be used for packages from new EGEE SA1 repo (**grid-monitoring-config-gen 0.10.0-1** and newer).

Important: Proxy refreshing cronjob is not generated by NCG. Cronjob is generated and managed by package `nagios-proxy-refresh`. Package requires separate manual configuration which is in case of Yaim-based installation generated by Yaim module.

NRPE installation: in case when NRPE UI is used, host certificate should be installed on NRPE UI server instead of Nagios.

- Find appropriate MyProxy server
  - MyProxy server should be configured as described below or apply patch as described here
  - In case of NRPE UI /C=XX/O=XX/OU=XX/CN=host/nagios.example.org is DN of NRPE UI host certificate.
- Install host certificate on Nagios server. In case of NRPE UI install host certificate on NRPE UI server.
  - See procedure below.
- Install user credential (same as below)
- On the Nagios or NRPE UI server
  - Configure nagios-proxy-refresh `/etc/nagios-proxy-refresh.conf` manually:
    `MYPROXY_HOST=<MyProxyServer>`
  - Other options are set to default, change only if necessary.
  - Start nagios-proxy-refresh service:
    `/etc/init.d/nagios-proxy-refresh start`

Old procedure

This procedure should be used for package **grid-monitoring-config-gen 0.9.11** and newer.

Important: new type of MyProxy credential requires gLite UI 3.1 because version of MyProxy deployed with gLite 3.0 doesn't support needed options.

Local probes which are executed by Nagios use short-term (12hr) proxy credential. This short-term credential is renewed periodically from a longer term proxy lodged in a trusted MyProxy server and protected by host certificate of Nagios server. The appropriate VOMS service is automatically contacted on renewal to decorate the proxy credential with VO attributes.

Important change in new procedure is that Nagios server now requires host certificate. In old procedure MyProxy was protected only with passphrase which allows potential attacker to retrieve it with brute force attack. By protecting it by host certificate, higher level of security is achieved.

- Find appropriate MyProxy server
  - MyProxy server should contain following line in `/etc/myproxy-server.config`:
    `authorized_retrievers "*"`
    `trusted_retrievers "*"`
    or:
    `authorized_retrievers "/C=XX/O=XX/OU=XX/CN=host/nagios.example.org"`
    `trusted_retrievers "/C=XX/O=XX/OU=XX/CN=host/nagios.example.org"`
    where /C=XX/O=XX/OU=XX/CN=host/nagios.example.org is DN of Nagios host certificate.

- **Warning** On gLite MyProxy nodes file `/etc/myproxy-server.config` is regenerated each time service `myproxy` is restarted. In order to add line of configuration one needs to add following to the script `/etc/init.d/myproxy-generate-config.pl`:
  ```
  $config .="authorized_retrievers \"*\"\n";
  $config .="trusted_retrievers \"*\"\n";
  ```
  or:
  ```
  $config .="authorized_retrievers
  \"/C=XX/O=XX/OU=XX/CN=host/nagios.example.org\"\n";
  $config .="trusted_retrievers
  \"/C=XX/O=XX/OU=XX/CN=host/nagios.example.org\"\n";
  ```
  where /C=XX/O=XX/OU=XX/CN=host/nagios.example.org is DN of Nagios host certificate. This line should be added before last section of script where $config is printed to config file (e.g. to line 50).

- Install host certificate on Nagios server
  - Certificate should be installed on standard location:
    ```
    /etc/grid-security/hostcert.pem
    /etc/grid-security/hostkey.pem
    ```
  - Fetch host certificate DN, which is needed for installing user credential:
    ```
    openssl x509 -in /etc/grid-security/hostcert.pem  -noout -subject
    ```

- Install user credential
  - Log in on a **trusted machine** as the user whose credential will be used to run the local service probes.
  - Enforce creating old style MyProxy by setting following environment variable:
    ```
    export GT_PROXY_MODE="old"
    ```
  - Store into a trusted MyProxy server a credential valid for an extended period. e.g 2 weeks -
    ```
    myproxy-init -c 336 -k NagiosRetrieve -s <MyProxyServer> -l nagios -s
    myproxy.example.org -x -Z "/C=XX/O=XX/OU=XX/CN=host/nagios.example.org"
    ```
  - **This credential will have to be renewed manually before the validity period is over.** See also User Unknown troubleshooting error below if using the gLite 3.1 UI.

- On the Nagios server
  - Check proxy refresh by executing cron job command generated by NCG and stored to file `/etc/cron.d/nagios_proxy_refresh` as user root.

Old procedure (2)

This procedure should be used for versions of package **grid-monitoring-config-gen** older than **0.9.11**.

Warning! Probe refresh_proxy performs VOMS proxy reordering hack in order to solve VOMS issue. In newer versions of gLite UI VOMS issue is resolved so the probe is generating incorrect proxies. Incorrect proxy causes issues for all probes which perform GridFTP transfers (hr.srce.GridFTP-Transfer, hr.srce.CAdist-Version, hr.srce.GRAM-Command, hr.srce.WMProxy-RunJob). Interim solution is to comment out following lines (353,354) in file /opt/lcg/share/grid-monitoring/probes/hr.srce/refresh_proxy: ($state,$tmpAnswer,$res) = reorderVomsProxy($proxy); $answer .= $tmpAnswer if ($tmpAnswer); More details can be found on following Savannah bug.

The security model implemented uses a short-term (12hr) proxy credential to launch the tests from the Nagios service. This short-term credential is renewed periodically from a longer term proxy lodged in a trusted MyProxy server and protected by a passphrase. The passphrase is stored in a clear-text file on the local Nagios service. The appropriate VOMS service is automatically contacted on renewal to decorate the proxy credential with VO attributes.

- Install user credential

- Logged in on a **trusted machine** as the user whose credential will be used to run the local service probes :
  Store into a trusted MyProxy server a credential valid for an extended period. e.g 2 weeks - `myproxy-init -c 336 -k NagiosRetrieve -s <MyProxyServer> -l nagios`
  This credential will have to be renewed manually before the validity period is over. See also User Unknown troubleshooting error below if using the gLite 3.1 UI.

- On the Nagios server
  - Store MyProxy proxy certificate passphrase (e.g. in `/etc/nagios/globus/proxy_passphrase`) Check that access to this file is restricted to user `nagios`.
  - Check proxy refresh check: (only if running local probes)
    ```
    su - nagios -c "/opt/lcg/share/grid-monitoring/probes/hr.srce/refresh_proxy
    -u <MyProxyServer>  -m hr.srce.GridProxy-Get --name NagiosRetrieve
    --passfile /etc/nagios/globus/proxy_passphrase --vo dteam
    --proxy=/etc/nagios/globus/userproxy.pem -n"
    ```
    Which should report `MyProxy credential retrieved. VOMS credential retrieved. VOMS proxy reordered.`

NCG Configuration

Run WLCG Nagios configuration generator:

```
ncg.pl
```

By default the output is stored as a set of Nagios configuration files created in the directory `/etc/nagios/wlcg.d`.

Important: each time site configuration changes (e.g. new services are added, hosts are removed) it is necessary to rerun `ncg.pl` and restart nagios service (`/etc/init.d/nagios restart`).

Input to the configurator is taken by default from the file `/etc/ncg/ncg.conf` which must be edited by the site administrator to reflect the desired input. The location of the config input file can be changed using the --config option. To view valid all options run: `ncg.pl --help`. You will at least have to specify your site name as `SITENAME`. The following paragraphs describe the format of the configuration file and how to access information about the valid input parameters.

Config file `ncg.conf` uses the same structure as *Apache HTTP Server* configuration and thus provides for the setting of 'global' and module-specific parameters according to markup sections which correspond to the perl modules inside NCG. A little knowledge of perl does help in understanding the structure of the input but not a prerequisite. For example, consider the following snippet taken from `ncg.conf` (note this does not define a complete configuration):

```
GLITE_VERSION=3.1.0
<NCG::ConfigGen>
  <Nagios>
    MYPROXY_SERVER=${MYPROXY_SERVER}
    PROBES_TYPE=remote
  </Nagios>
</NCG::ConfigGen>
```

In the above snippet, the global parameter GLITE_VERSION is given the value '3.1.0'. Other parameters are specified for the module NCG::ConfigGen::Nagios with the value for MYPROXY_SERVER being taken from the same-named environment variable. Each module has a separate section, allowing for a flexible and modular configuration. Example configurations are included in the distribution in the directory `/etc/ncg` and are generally self documenting.

In addition to the global section the following module sections are defined:

- Topology -
  - NCG::SiteInfo - controls the gathering of information describing a Site's hosts and services
  - NCG::LocalRules - controls the local manipulation of the configuration by the addition or removal of hosts, services, contact information etc.
- Probe Description (it is not expected that the average user using default installations should have to change or configure probe descriptions) -
  - NCG::LocalMetrics - defines metrics in terms of probe to be used, attributes to pass to the probe, VO and other metric dependencies
  - NCG::LocalMetricSets - controls which sets of metrics are appropriate to test a specific Grid service (sometimes called node type). In effect this defines a grouping of *sub-services*.
  - NCG::LocalMetricsAttrs - controls the gathering of variable metric attributes (e.g. actual service port number used) from information sources such as the information system or by applying service specific heuristics.
  - NCG::RemoteMetrics - controls the extraction of the lists of remote (off-site/central services which probe the site services) metrics available for the site services.
- Configuration Generation -
  - NCG::ConfigGen - controls the final phase of configuration generation for a specific monitoring tool

The NCG configurator is written in *perl* and each module is self-documenting. Additional information describing available keyword parameters and local file formats can be found by using the *perldoc* utility.

1. Example to see information about the file format for defining local metrics use the command `perldoc NCG::LocalMetrics::File`
2. Example to see information about generating configuration for Nagios use the command `perldoc NCG::ConfigGen::Nagios`

Specifying a full path in the examples is also possible. e.g. `perldoc /usr/lib/perl/vendor_perl/5.8.5/NCG/LocalMetrics/File.pm`.

Look in GridMonitoringNcgRecipes for examples on using NCG configuration.

By default for Nagios, NCG generates the following configuration files in `/etc/nagios/wlcg.d` : `commands.cfg contacts.cfg hosts.cfg services.cfg wlcg.nagios.cfg`. Where an existing Nagios installation exists and local configuration manipulation via `/etc/ncg/ncg.conf` or referenced files is insufficient to allow proper integration of the Grid service monitoring without changes the separation of the output configuration more easily allows the possibility of further local customisation.

NRPE UI configuration

In case when NRPE UI is used modify NCG::ConfigGen::Nagios section of `/etc/ncg/ncg.conf`:

```
<NCG::ConfigGen>
  <Nagios>
    ...
    NRPE_UI=<NRPE_UI_SERVER>
  </Nagios>
</NCG::ConfigGen>
```

Configuration file for NRPE service will be generated in directory `/etc/nagios/nrpe`. Site administrator must manually transfer this directory to NRPE UI server.

Multi site configuration

NCG starting from version 0.9.12-0 supports generating configuration for monitoring multiple sites with a single Nagios instance. Currently the only supported mechanism is to list sites in a static file.

- Add NCG::SiteSet::File section to `/etc/ncg/ncg.conf`:

```
<NCG::SiteSet>
  <File>
    DB_FILE=<LOCAL_FILE_CONFIG>
  </File>
</NCG::SiteSet>
```

- Comment global definition of variable `BDII` in file `/etc/ncg/ncg.conf`. This is required because of the following bug.

- Add list of sites to file :
  - ♦ If site is present in SAM (module NCG::SiteInfo::SAM is included):

```
SITE!<sitename1>
SITE!<sitename2>
...
```

- ♦ If site is not present in SAM (only module NCG::SiteInfo::LDAP is used):

```
SITE_BDII!<sitename1>!<site1_bdii>
SITE_BDII!<sitename2>!<site2_bdii>
...
```

- In case that sites are not present in SAM make sure that `ADD_HOSTS` variable in NCG::SiteInfo::LDAP is switched on:

```
<NCG::SiteInfo>
  <LDAP>
    ...
    ADD_HOSTS=1
  </LDAP>
</NCG::SiteInfo>
```

- In case you had an existing single site installation manually remove config directory `/etc/nagios/wlcg.d`.

- Rerun `ncg.pl`

Nagios Configuration

Configure Nagios service

Make the following changes to `/etc/nagios/nagios.cfg` to include WLCG configuration into the base Nagios configuration.

- Comment out the line:
  `cfg_file=/etc/nagios/localhost.cfg`
- Add the following line:
  `cfg_dir=/etc/nagios/wlcg.d`
- If you configured for remote probes (with or without local probes)

- Set the following values
  `check_external_commands=1`
  `log_passive_checks=0` (optional)

- If you configured for local probes
  - Add the following line
    `resource_file=/etc/nagios/wlcg_resource.cfg`
  - Set the following values
    `service_check_timeout=900`

- Check Nagios configuration
  `nagios -v /etc/nagios/nagios.cfg`

- If everything is OK start Nagios service
  `service /nagios start`

Configure the nagios Web Interface

Nagios restricts users' rights to view status and schedule commands at the Web interface via configuration in `/etc/nagios/cgi.cfg`. A restrictive access profile is configured by default and you must define users and set their rights as described below. You can use traditional password-based authentication or, if your users have appropriately trusted certificates loaded in their browsers, you can choose to use certificate-based authentication.

- Password-based authentication
  - Create a password file and generate user `nagiosadmin` using the command -
    `htpasswd -c /etc/nagios/htpasswd.users nagiosadmin`
    Additional users can be added by running the same command omitting the `-c` option. See `htpasswd -h` for more options or refer to the Nagios documentation.
  - Option `SSLRequireSSL` can be added (Apache module mod_ssl required) to the directory configurations in `/etc/httpd/conf.d/nagios.conf` to ensure that passwords are only used over an encrypted link. Optionally for convenience of your users the following configuration can also be included to redirect users who connect via unencrypted http.
    ```
    Options +FollowSymLinks
    RewriteEngine on
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
    ```
    The rewite rules must be added at the server or virtual host configuration level to be applied before the authentication phase.
- Certificate-based authentication
  - Full details of configuring SSL for Apache can be found in the mod_ssl documentation but the following basic steps act as a quick-start guide. The following assumes that the certificates of the trusted Certificate Authorities are loaded in to `/etc/grid-security/certificates` and the certificate and corresponding private key for the host are in `/etc/grid-security/hostcert.pem` and `/etc/grid-security/hostkey.pem` respectively, which is the most common configuration for the gLite UI. Edit the file `/etc/httpd/conf.d/ssl.conf` and change or uncomment lines to set the values shown below.
    ```
    SSLCertificateFile /etc/grid-security/hostcert.pem
    SSLCertificateKeyFile /etc/grid-security/hostkey.pem
    SSLCACertificatePath /etc/grid-security/certificates
    SSLCARevocationPath /etc/grid-security/certificates
    ```
  - Adding the following lines to the directory configurations in `/etc/httpd/conf.d/nagios.conf` requires users to authenticate to the server using their certificate -

```
SSLRequireSSL
SSLOptions +FakeBasicAuth
SSLVerifyClient require
SSLVerifyDepth 3
```

- Add each user's certificate Subject DN to `/etc/nagios/htpasswd.users` repeating the exact password string used in the example below -
  ```
  /DC=org/O=myplace/OU=Users/CN=My Name:xxj31ZMTZzkVA
  /DC=org/O=anotherplace/OU=Users/CN=My Friend:xxj31ZMTZzkVA
  ```
- If mod_ssl has been configured to check certificate revocation lists (CRLs) (recommended) then these must be updated periodically. In the standard gLite-UI configuration CRLs are updated using a `fetch-crl` cron and Apache must be restarted periodically to load the updated information. Creating the file `/etc/cron.d/apache-restart` with the following contents will do this (times can be adjusted to optimally do this after the `fetch-crl` cron has run).
  ```
  PATH=/sbin:/bin:/usr/sbin:/usr/bin
  30 11,17,23,5 * * * root /usr/sbin/apachectl graceful >> /dev/null 2>&1
  ```

After users are properly authenticated you can give them appropriate permissions as indicated in `/etc/nagios/cgi.cfg`. For certificate-based authentication the username is replaced by the certificate Subject DN. For example -
`authorized_for_all_services=nagiosadmin,/DC=org/O=myplace/OU=Users/CN=My Name,/DC=org/O=anotherplace/OU=Users/CN=My Friend`
To manually schedule checks and issue other commands from the Web interface user names or Subject DNs should be to the entries `authorized_for_all_service_commands` and `authorized_for_all_host_commands`.

Start the web server - `service httpd start`

Configure NRPE service

In case when NRPE UI is used, site administrator needs to configure NRPE service:

- Transfer configuration generated on Nagios server by NCG (directory `/etc/nagios/nrpe`) to NRPE UI server manually
- Add the following line to `/etc/nagios/nrpe.cfg`:
  `=include_dir=/etc/nagios/nrpe/`
- Start NRPE service
  `/etc/init.d/nrpe start`

Notice: in case of Yaim-based installation glite-NRPE module will provide automatic mechanism for synchronizing directory `/etc/nagios/nrpe` between Nagios and NRPE UI server via HTTPS.

Using Nagios

You should now have a working Nagios Installation and can go to http://<host>/nagios/ with your browser to view the service status. Please check the Troubeshooting section below if you have problems.

Initial checks and service status may take a few minutes to be completed and displayed so be patient. Note that email and other notifications by Nagios about service problems are switched off by default when the configuration is generated. These can be enabled by editing the 'define contact' configuration in (by default) /etc/nagios/wlcg.cfg. Any changes you make to this file will currently be overwritten and lost if the WLCG Nagios configuration generator is re-run so make sure to make a copy of your changes.

Information on using Nagios and standard procedures for management of checks and alerts is available in the

documentation from the Central European ROC of the EGEE Project and the Nagios documentation (also available from the Nagios service web interface).

Host and service groups

NCG groups hosts and services based on several criteria.

Hosts are grouped based on:

- service role (e.g. CE, SE, VOMS, LFC)
- site membership (all nodes with services are members of site hostgroup).

Services (checks) groups are following:

- metric set to which metric belong (e.g. globus-GRAM, gsiftp, MyProxy); metric set names are defined by WLCG probes (local probes only)
- remote monitoring service from which result was gathered (sam, npm) (remote probes only)
- VO for which probe was executed (local and remote probes which are VO-dependent)
- probe types (local, remote, native)
- nagios-internal groups which contains only internal checks (e.g. GridProxy-Valid, SAM-Gather, NPM-Gather, ...).

Firefox plugin for Nagios

There is a very useful Nagios plugin for Firefox that enables users to get instant notifications from Firefox's status bar in case of failure on site.

Installation is fairly easy:

- If you configured Nagios web interface to user certificate-based authentication, import valid user certificate into Firefox.
- Install plugin from following address (installation requires Firefox restart).
- After Firefox restart right click "N" icon on the right side of status bar and select Settings. On tab "General" select "Add new". Fill following fields:
  Name: (e.g.) EGEE
  Nagios web interface url: http://<host>/nagios/
  click "Locate url" and if field doesn't get filled, switch on checkbox "Set url manually", and set Location url fields to: https://<host>/nagios/cgi-bin/status.cgi?style=detail
- If you configured Nagios web interface to use username/password authentication, set username and password fields.
- Close Settings window and you should see little green textbox "No problem".

Further tuning is recommended:

- On tab Filtering switch on all checkboxes between "Host and services that have been acknowledged" and "Services on down or unreachable and acknowledged hosts".
- On tab display Behaviour set Blinking to "Blink all types if a new problem appears" or "Off".
- On tab Sounds set Sounds to "Off" (unless you'd like to hear it shout when each new problem appears).

Concerning using the status bar if a problem appear:

- by hovering over it you'll get pop-up window with detailed information about hosts and services in problem state

- by selecting Go to Services you'll get to Nagios page with detailed information about all problems, e.g.: https://<host>/nagios/cgi-bin/status.cgi?style=detail&servicestatustypes=28&hoststatustypes=15.

Troubleshooting

In this section we collect common problems and solution recipies which may be useful for your configuration. For general Nagios issues you might find the answer in the Nagios FAQ. Please also check in the Group Savannah bug tracker to see if your problem is listed there.

- Clicking anything on the left-side Nagios page reports an `Internal Server Error`
  Check in `/var/log/httpd/error_log`. If you see messages relating to permission denied for exec on .cgi programs then this can be caused when **SELinux** is not configured to allow Apache to execute the Nagios programs. You can (for RHEL4 and derived systems) follow these instructions from Red Hat or disable SELinux in `/etc/selinux/config` and reboot. The author (who **disclaims** any responsibility that this is fit for any purpose) managed to combine the above guide with this mail-thread and found that, without disabling SELinux, the configuration could be made functional (tested on SLC4) by -
  - ♦ install the SELinux targeted policy sources (`selinux-policy-targeted-sources`)
  - ♦ put the following content in
    `/etc/selinux/targeted/src/policy/domains/misc/local.te` -
    ```
    allow httpd_sys_script_t var_log_t:dir search;
    allow httpd_sys_script_t var_log_t:file { getattr read };
    allow httpd_sys_script_t var_log_t:fifo_file { getattr write };
    ```
  - ♦ `cd` to `etc/selinux/targeted/src/policy` and execute `make load`
  - ♦ change the security context on the Nagios cgi's by running -
    ```
    chcon --reference=/var/www/cgi-bin -R /usr/lib/nagios/cgi
    ```

- The metric `org.wlcg.aggregate-status` is always pending for all hosts.
  The script for this metric (which is calculated as a event handler triggered on any state change of any metric) relies on the python library `optparse`. This is provided by default in python2.3, but not in python2.2. For python2.2 you must install the additional rpm 'python-optik' which provides the optparse library.

- `Error: ...User unknown to this VO` from hr.srce.GridProxy-Get
  At time of writing the gLite 3.1 distribution has a problem that the UI myproxy-init command generates a "new-style" rfc-compliant proxy credential on the myproxy server. However, voms-proxy-init with the -noregen option, as used to generate credentials to run monitoring probes, is unable to authenticate with the voms service using this credential and generates and "Error: ... User unknown to this VO" error message. Consequently, if using gLite 3.1 as a platform you should either use a 3.0 UI for delegating your credential to myproxy or set the GT_PROXY_MODE to "old". More details in GGUS ticket 26846.

- The myproxy server used to store authentication credentials stops accepting delegation requests. /var/log/messages on the server contains messages of the form: `myproxy-server: Error authenticating client: GSS Major Status: Some Other GSS failure GSS Minor Status Error Chain: (null)Error reading token` corresponding to the times of failed attempts by Nagios to renew credentials.
  This is a known problem with more details and a workaround reported here.

- The myproxy server used to store authentication credentials stops accepting delegation requests. /var/log/messages on the server contains messages of the form: `myproxy-server: Error authenticating client: GSS Major Status: Some Other GSS failure GSS Minor Status Error Chain: (null)Error reading token` corresponding to the times of failed attempts by Nagios to renew credentials.
  This is a known problem with more details and a workaround reported here.

- After upgrading from older versions verifying Nagios config returns following error:
  ```
  nagios -v /etc/nagios/nagios.cfg
  Error: Cannot open resource file '/etc/nagios/wlcg_resource_remote.cfg' for reading!
  ```
  Beginning with grid-monitoring-config-gen-0.9.11-0 this file is not used anymore. Removing following line from `/etc/nagios/nagios.cfg` will solve problem:
  ```
  resource_file=/etc/nagios/wlcg_resource_remote.cfg
  ```

This topic: LCG > GridMonitoringNcg
Topic revision: r20 - 2008-11-19 - JamesCasey