Table of Contents

SAM - Nagios for ATLAS

SAM NAGIOS for ATLAS details

Configs

On sam-atlas-dev the procedure for installing the gfal2 python APIs breaks down as follows:

- `yum install gfal2-python`
- `yum install gfal2-plugin-srm gfal2-plugin-gridftp`
- `yum install gfal2-plugin-srm gfal2-plugin-xrootd`
- `yum update gfal2-*`
- `yum --disablerepo="*" --enablerepo="lcgutil-cbuilds-el5" reinstall gfal2-plugin-srm`

The last step is meant for installing new plugin versions which don't figure yet in the main release. Once the new features you need are introduced in a stable release it should not be needed.

In order to let yum to search into lcgutil repos, add the file `/etc/yum.repos.d/lcgutil-cbuilt.repo` with the follwing content (assuming you're working on a SL6 machine and substituting `$basearch` with the machine arch):

```
[lcgutil-cbuilds-el6]
name=LCGUTIL Continuous Build Repository
baseurl=http://grid-deployment.web.cern.ch/grid-deployment/dms/lcgutil/repos/el6/$basearch
gpgcheck=0
enabled=1
protect=1
```

The gfal2 developers to be contacted if some help or feedback is needed for instruction and/or information about libraries in production repositories are alejandro.alvarez.ayllon@cernNOSPAMPLEASE.ch and adrien.devresse@cernNOSPAMPLEASE.ch.

SRM-Probe

Get DDM Endpoints json file from AGIS

The only input file for the tests is a json containing configuration information for all the ATLAS DDM endpoints. A daily script tries to collect it from here. Should the download fail, the dictionary necessary for executing tests on SE is cached from previous executions.

Outline of new tests

For each test (except for *GetATLASInfo*) if an error is returned from the gfal2 API or the timeout limit is infringed, the test outcome is assigned as CRITICAL.

- *LsDir*: for each directory associated to a site it uses gfal2 `listdir` API to list the first 10 elements of the directory. If it fails on all the tokens, all permissions are blacklisted for the site SE.
- *Put* and *Del*: a newly created low size file is attempted to be copied onto the directories with `copyfile` API. If the copy is successful the deletion of the same file follows via `unlink` method. The same code is used both for *Put* and *Del* tests. If the file copy fails the *Del* exit status is WARNING.
- *Get*: the test makes use of the `copyfile` API with a suited syntax to perform the 'get' action of a fixed-name file. A preliminary `stat` method is executed to check for the file exists. If it doesn't it is copied again before going through with the *Get* test. A WARNING exit status is issued if any of the preliminary checks fail.

- Each of the *Put*, *Get*, *Del* test is executed also on each single token at time.

`*TODO*`

- Implement PrepareToGet and PrepareToGet tests

Such configuration makes only the *Put* and *Del* test inter-dependent on each other, while any other test can be executed independently, as long as *GetATLASInfo* has been already executed.

Migration of SAM probes from SAME based to Nagios based infrastructure

All the technical details have been recorded here PracticalHints, sharing the sam2nagios migration experience with the other VOs.

Building RPMs for Nagios

Although the suggested method to build RPMs is Koji (as explained in the forementioned twiki page), most of the time it's easier and faster to build the RPMs from a lxplus node.

Following are the instructions to build a RPM on lxplus from a local development area.

What you need

- Get the source code under a local directory (in the following instructions, the directory containing the source is supposed to be called `org.atlas`); it can be downloaded with

```
svn checkout https://www.sysadmin.hep.ac.uk/svn/grid-monitoring/trunk/probe/org.atlas org.atlas
```

- the *spec* file that is used by the *rpmbuild* command (an example is on AFS at *~gnegri/public/nagios/grid-monitoring-probes-org.atlas.spec*)
- (optional!) a bash script that calls the building and performs a few checks (an example is on AFS at *~gnegri/public/nagios/makeRPM.sh*)
- a *.rpmmacros* file in your `$HOME` directory made like this (use your AFS username whenever needed):

```
%_topdir        /tmp/gnegri
%_dbpath        /tmp/gnegri/db
%_tmppath       /tmp/gnegri/rpm-tmp
%_sourcedir /afs/cern.ch/user/g/gnegri/public/nagios/SOURCES
```

What to edit

- in *grid-monitoring-probes-org.atlas.spec*: change the `Version` number and, if needed, the `Release`
- in *makeRPM.sh*: in principle, nothing should be changed

What to check before building

- Notice that n the *spec* file the building is done using a *buildroot*. This *buildroot* is constructed starting from the *_tmppath* defined in the *.rpmmacros* in your `$HOME` directory
- If you're using the `makeRPM.sh` script to run the building, notice that it has to be called with the release number and it has to be the same release number as defined in the *spec* file

Building the RPM

Supposing you're using the `makeRPM.sh` script, simply launch it as

```
sh makeRPM.sh <Version_number>
```

Installing RPMs for Nagios

RPMs for Nagios have to be uploaded to the EGEE software reository at CERN and the Quattor profile of the Nagios server has to be edited and updated.

In order to do these actions, you have first to ask for access to:

- lxvoadm (ask for membership to the *LxVoAdm-ATLAS* e-group at https://e-groups.cern.ch/
- CDB (ask for it from SNOW asking for *Service: cdbserv.cern.ch* and *CDB acl group: %gridsam_atlas*)
- swrepsrv (ask for it from SNOW asking for *Service/cluster: Nagios/SAM* and *area: /egee/glite*)

Before connecting to CDB, you have to edit the *.cdbop.conf* file in your home:

```
vi ${HOME}/.cdbop.conf
#-----------------------------------------------
protocol = https
server = cdbserv.cern.ch
#-----------------------------------------------
```

Similarly, connection to *swrepsrv* requires a configuration file in your home directory:

```
vi ${HOME}/.swrep-soap-client.conf#
#-----------------------------------------------
# Repository location (Address of SWRep Server)
server = swrepsrv.cern.ch
# Timeout for SOAP connection
# timeout = 7200
#-----------------------------------------------------------------------
# * debug level (1 to 5)
# debug =
# * verbose output
# verbose =
#-----------------------------------------------
```

You can now upload your RPM to the /egee/glite repository: log into lxvoadm and execute

```
swrep-soap-client put x86_64_slc5 /egee/glite <local_path_to_RPM>
```

Now, connect to CDB (from lxvoadm) to get/edit/upload the quattor template:

- `cdbop` will open an interactive session on CDB
- `get profiles/profile_samnag013.tpl` will retrieve the profile of the machine you want to update
- `!vi profiles/profile_samnag013.tpl` (notice the leading "!" that converts the call to a shell command) will edit the the just downloaded profile
- edit the line

```
/software/packages" = pkg_repl("grid-monitoring-probes-org.atlas","0.0.8-2","noarch");
```

- `update profiles/profile_samnag013.tpl` to update the profile in the Quattor repository
- `commit -f -c "<comment>"` to force the update of the profile on the server

You can verify that the new profile/!RPM doesn't clash with the rest of the installation by logging into the just updated machine as root and execute:

```
spma_wrapper.sh --noaction
```

Insert new CE tests

The procedure consists in modifying three files and restarting the ncg configuration script. The break down is:

- Modify the file `/usr/libexec/grid-monitoring/probes/org.atlas/wnjob/org.atlas/etc/wn.d/org.atlas/service.c` by inserting a new `define service` block for the new test. The values of the three fields have to be set as:
  - `use`: the SAM-Nagios profile the test has to be run within (e.g.: `use sam-generic-wn-active`);
  - `service_description`: the test name as it is going to be displayed by Nagios interface (e.g.: `service_description org.atlas.WN-cvmfs-`);
  - `check_command`: the command to be executed by the Nagios box for running the test (e.g.: `check_command CE-ATLAS-WN-cvmfs`).
- Modify the file `commands.cfg` in the same directory as the previous bullet by inserting a new `define command` block where the fields values have to be set as:
  - `command_name`: has to replicate the value in the field `check_command` in the file `service.cfg`;
  - `command_line`: the command to be actually executed from the command line (e.g.: `command_line $USER3$/org.atlas/CE-ATLAS-WN-cvmfs`).
- Modify the file `/etc/ncg-metric-config.d/atlas.conf` and insert a new block referring to the new test (names following to `service_description` field in `service.cfg`) with suited values for the contained field, e.g.:

```
"org.atlas.WN-cvmfs" : {

"parent" : "emi.ce.CREAMCE-JobState",

"flags" : {

"OBSESS" : 1,

"VO" : 1,

"PASSIVE" : 1

},

"metricset" : "org.atlas.WN"

},
```

- Finally refresh the executed tests by executing the script `ncg.reload.sh`.

SAM machines for ATLAS

alias	real name	notes
sam-atlas-prod	samnag035	
sam-atlas-preprod	samnag041	
sam-atlas	samnag013	to be discarded

-- AleDiGGi - 20-Jul-2010

This topic: LCG > SAMToNagiosATLAS
Topic revision: r24 - 2015-06-23 - SalvatoreAlessandroTupputi