INSTALLATION OF A DASHBOARD REGIONAL INSTANCE

Introduction

This presentation will propose you an installation of the dashboard in a regional context on a Scientific Linux 5 system.

We will propose today a test package.

To avoid any trouble with central dashboard, we will install a limited regional instance with Nagios notification system.

Consequently we will an installation with :

- a Nagios notification system
- a connection to a test instance of Ggus
- communications by emails sent to portal webmaster with a "test" flag in the subject

A) Installing and setting up Lavoisier web service

1. Deploy Lavoisier

```
tar -xzf lavoisier.tar.gz
```

Nb :

- you can set up an environment variable refering to your Lavoisier home directory (called LAVOISIER_INSTALL_DIR in this document)
- in the future you could use a rpm package by using this command : rpm -ivh lavoisier.1-1.src.rpm

2. Configure the web service and the notifications corresponding to your "ROC"

Prequisites : a certificate server for Lavoisier service !

Location : LAVOISIER_INSTALL_DIR/etc

Edit Lavoisier-config.properties file

=> Set database connection (the same that you should fill in point B.2 !)

#PARAMETERS RELATED to NAGIOS / ACTIVE MQ / YOUR DB HOSTING THIS INFORMATION
nagios.url=tcp://gridmsg002.cern.ch:6166 **<= don't change this line !**
nagios.database.url=jdbc:mysql://your_db_host/your_db_name **<= replace host and db**

```
nagios.database.username=username <= replace it
nagios.database.password=password <= replace it
nagios.table=NOTIFICATION <= don't change this line !
```

=> Set path to your server certificate

```
certificate.path=/opt/lavoisier/certificate/lavoisier.p12
```

=> Uncomment the line corresponding to your ROC in the list:

```
# PARAMETERS RELATED TO REGIONAL INSTANCE
#roc.name=AsiaPacific
#roc.name=Central Europe
#roc.name=CERN
#roc.name=France
#roc.name=Italy
#roc.name=GermanySwitzerland
#roc.name=NorthernEurope
#roc.name=ROC_Canada
#roc.name=ROC_IGALC
#roc.name=ROC_LA
#roc.name=SouthEasternEurope
#roc.name=SouthWesternEurope
roc.name=UKI <= here for example !
```

3. Start the service

Prerequisites : 8000 and 9000 ports have to be opened

Location : LAVOISIER_INSTALL_DIR/

Execute : sh lavoisier-restart.sh

This script stops (if service is up) and starts the service

4. Access to the monitoring web interface

Once the service is up (~ 1min.) you can access to monitoring web pages

Interface showing the different views with their parameters and configuration
http://lavoisierhostname:9000/LavoisierService/view/index

Interface showing the status and the date of refreshing of views
http://lavoisierhostname:9000/LavoisierService/view/status

Config file in line
http://lavoisierhostname:9000/LavoisierService/view/config

Graph of dependencies of the views
http://lavoisierhostname:9000/LavoisierService/view/dependencies

For more informations see :
https://forge.in2p3.fr/attachments/188/lavoisier_documentation.doc

B) Installing and setting up the PHP web application

The Dashboard tool for Grid Operators is implemented as a web-based application, it is written in the PHP programming language, and uses the Symfony <http://www.symfony-project.org/> templating system. The

Symfony framework files are bundled with the application, so you don't need to install Symfony separately.

---

1. Deploy the source code

```
tar –xzf operations-portal-1.1.tar.gz
```

Nb :
- you can set up an environment variable refering to your web home directory (called INSTALL_DIR in this document)
- in the future you could use a rpm package by using this command : rpm -ivh operations-portal-1.1-1.src.rpm

Tips: you can check that your web environment is ready by running the php script check_configuration.php (in INSTALL_DIR/web)
(see http://www.symfony-project.org/getting-started/1_2/en/02-Prerequisites)

2. Configure database connection

Location : INSTALL_DIR/

For example we are going to use a Mysql database **(the same that you should fill in point A.2 !)**

Host : ccmysql.in2p3.fr
Database name : cic_france
User : cic_france
Password : Q51N4Vk

Replace parameters with yours and execute this command :

```
symfony configure:database --name=dashboardConnection
--class=sfDoctrineDatabase "mysql:host=ccmysql.in2p3.fr;dbname=cic_france"
cic_france Q51N4Vk
```

NB : this command fills INSTALL_DIR/config/database.yml with the connection string passed (please use the command instead of edit this file)

3. Build database

Location : INSTALL_DIR

Execute this command :

```
symfony doctrine:build-all
```

NB :
This command create dynamically tables in the database (check that the database has 12 empty tables)
Point 2 and 3 have to be done sequentially

4. Configure filters and settings

These files could be used to set specific security and environment parameters. For this presentation, we will used default files.

Location : INSTALL_DIR/apps/frontend/config/

Rename *filters.yml.template* into *filters.yml*

Rename *settings.yml.template* into *settings.yml*

5. Configure app.yml

This file contains all variables and parameters used by the web application (in particular web services adresses)

Location : INSTALL_DIR/apps/frontend/config/

Rename *app.yml.template* into *app.yml*

Edit *app.yml* file

IMPORTANT LINES :

```
webconstants:

    # Title of your site in window top bar
    title: Regional CIC Portal Instance - France

    # Title of your site in web application header
    header_title: EGEE French Dashboard

    # Form security : unique secret to enable CSRF protection or false to disable
    csrf_secret: false
```

```
webservices:

        # Lavoisier webservice PROD (replace host)
        lavoisierprodurl: "http://ccgridvmli16.in2p3.fr:8000/LavoisierService?WSDL"
        lavoisiertesturl: "http://ccgridvmli16.in2p3.fr:8000/LavoisierService?WSDL"

        # Lavoisier CIC-GOCDB (replace host)
        lavoisiercicgocprodurl: "http://ccgridvmli16.in2p3.fr:9000/LavoisierService/view/"
        lavoisiercicgoctesturl: "http://ccgridvmli16.in2p3.fr:9000/LavoisierService/view/"
```

Tips :

- don't forget to open 8000 and 9000 ports on the machine who host Lavoisier web service

6. Clear symfony cache

This command is used to clear Symfony configuration files cache (here we want to clear app.yml and setting.yml caches)

Location : INSTALL_DIR/

Execute this command :

```
symfony cc
```

Tips:

- each time you make modifications on app.yml or settings.yml you have to clear Symfony cache
- From the moment that the web server is up and web alias is set, you can clear cache by invoke cc.php file (ex: http://operations-portal.in2p3.fr/cc.php) (you can rename this file, it is located in INSTALL_DIR/web/)

7. Create Apache alias for the website

All source code is hidden ; the web server has just to see the frontal controller (index.php) located in INSTALL_DIR/web

In our case the Alias definition looks like that (in *httpd.conf*) :

```
<VirtualHost *:80>
        ServerName ccgridvmli16.in2p3.fr
        DocumentRoot "/var/www/html/web"
        DirectoryIndex index.php
        Alias /sf /$operations-portal/web/sf
        <Directory "/var/www/html/web">
                AllowOverride All
                Allow from All
        </Directory>
```

```
</VirtualHost>
```

For more details see : http://www.symfony-project.org/book/1_2/03-Running-Symfony#chapter_03_sub_setting_up_a_virtual_host

8. GGUS Downtime

Location : INSTALL_DIR/apps/frontend/config/

Edit *app.yml* file

Change the variable *ggus* value to "*on*"

```
ggus: on
```

Clear cache (point B.6)

This configuration is to be used during a Ggus service downtime, it permits to disabled action with the ticketing system (tickets will be on a read-only mode)

9. SAM Downtime

Location : INSTALL_DIR/apps/frontend/config/

Edit *app.yml* file

Change the variable *sam* value to "*on*"

```
sam: on
```

Clear cache (point B.6)

This configuration is to be used during a SAM service downtime, it permits to disabled action with alarms system (alarms will be on a read-only mode)