



Project Acronym: **OPTIMIS**
Project Title: **Optimized Infrastructure Services**
Project Number: **257115**
Instrument: **Integrated Project**
Thematic Priority: **ICT-2009.1.2 – Internet of Services, Software and Virtualisation**

Monitoring Infrastructure Installation Guide

Activity 4: Basic Service Operation

WP 4.1: Monitoring and Data Collection

Due Date:	M36	
Submission Date:	07/06/2013	
Start Date of Project:	01/06/2010	
Duration of Project:	36 months	
Organisation Responsible for the Deliverable:	University of Stuttgart-HLRS (USTUTT)	
Version:	1.0	
Status	Final	
Author(s):	Pierre Gilet	USTUTT
	Tinghe Wang	USTUTT
	Daniel Espling	UMU
	Anthony Sulistio	USTUTT
Reviewer(s)	Wolfgang Ziegler	SCAI
	George Kousiouris	NTUA
	Karim Djemame	ULEEDS



Project co-funded by the European Commission within the Seventh Framework Programme

Dissemination Level

PU	Public	X
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group specified by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	



Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
1.0	2012-04-04	Created document.	Pierre Gilet
1.1	2012-09-24	Updated document.	Tinghe Wang
2.0	2012-01-24	Updated document for Y3.	Pierre Gilet, Tinghe Wang
2.1	2013-02-22	Updated document for Y3.	Pierre Gilet
2.2	2013-03-01	Updated section on Service Monitoring.	Daniel Espling
2.3	2013-03-04	Updated the document.	Pierre Gilet
2.4	2013-03-06	Updated the document.	Daniel Espling
2.5	2013-05-28	Added the OpenNebula and OpenStack sections.	Anthony Sulistio



Table of Contents

1	INTRODUCTION	5
1.1	GLOSSARY OF ACRONYMS.....	5
2	MONITORING INFRASTRUCTURE INSTALLATION GUIDE	6
2.1	PLATFORMS SUPPORTED	6
2.2	INSTALLATION INSTRUCTIONS	6
2.2.1	<i>MySQL Server and Client.....</i>	<i>6</i>
2.2.2	<i>Database Definition</i>	<i>6</i>
2.2.3	<i>Database Scripts and related Cronjobs.....</i>	<i>7</i>
2.2.4	<i>Nagios Server.....</i>	<i>7</i>
2.2.5	<i>Nagios Configuration and Plugins on Nagios Server</i>	<i>8</i>
2.2.6	<i>Plugins and NRPE Configuration on monitored physical Hosts.....</i>	<i>11</i>
2.2.7	<i>Apache Tomcat.....</i>	<i>13</i>
2.2.8	<i>Java RESTful Web Services</i>	<i>13</i>
2.2.9	<i>MySQL Client.....</i>	<i>14</i>
2.2.10	<i>RabbitMQ.....</i>	<i>14</i>
2.2.11	<i>Collector Scripts and related Cronjobs</i>	<i>14</i>
2.2.12	<i>OpenNebula Monitoring Plugin.....</i>	<i>14</i>
2.2.13	<i>OpenStack Monitoring Plugin</i>	<i>15</i>
2.2.14	<i>Service Level Monitoring and Probes</i>	<i>16</i>



1 Introduction

This document is the installation guide of the OPTIMIS Monitoring Infrastructure.

OPTIMIS aims towards Optimized Service Construction, Deployment, and Execution for Cloud Infrastructures by offering tools to efficiently manage the full life cycle of services. These tools provide simplified construction of services, help make informed deployment and runtime management decisions for evaluation of providers, and support Quality of Service provision.

This document assumes the reader is familiar with the OPTIMIS project vision and architecture. The following documents are recommended sources of information:

- Deliverable D1.2.1.2 OPTIMIS Architecture deliverable, October 2011
- OPTIMIS: a Holistic Approach to Cloud Service Provisioning. A. Juan Ferrer, F. Hernandez, J. Tordsson, E. Elmroth, C. Zsigri, R. Sirvent, J. Guitart, R.M. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, S.K. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forgo, T. Sharif, and C. Sheridan. Future Generation Computer Systems, Elsevier, Vol. 28, No. 1, pp. 66-77, 2012.

Details on the OPTIMIS project are found on <http://optimis-project.eu>

1.1 Glossary of Acronyms

Acronym	Definition
MI	Monitoring Infrastructure
IPVM	Infrastructure Provider's Virtual Machine
DBVM	Database Virtual Machine

2 Monitoring Infrastructure Installation Guide

2.1 Platforms Supported

Developed and tested on Linux CentOS, Debian and Ubuntu.

2.2 Installation Instructions

2.2.1 MySQL Server and Client

The MySQL database server and client are required. The MySQL database server will host the MI database storing all the monitored data. The MI database will also host stored procedures used for data maintenance.

The installation of the MySQL database server and client on Linux can be usually done via repositories. Example on CentOS:

```
yum install mysql-server
yum install php-mysql
yum install mysql
```

Once the database server is up and running, create a user called "mmanager_usr". Grant all privileges to that user.

Location of installation: DBVM.

MySQL server version: 5.1.65 or higher.

2.2.2 Database Definition

Download the MI_Scripts directory (and all its subdirectories) from the SVN repository to your local drive. Go to DBVM/Database_Definition. Execute the script named "optimis_db.sql" with mysql. Example on Linux:

```
mysql -h [database_host] -u mmanager_usr -p[password] <optimis_db.sql
```

Note that no space is allowed between "-p" and the password.

This will create the optimis_db database and its tables:

- infrastructure_provider

-
- monitoring_resource_energy
 - monitoring_resource_physical
 - monitoring_resource_service
 - monitoring_resource_virtual
 - physical_resource
 - virtual_resource

Location of installation: DBVM.

2.2.3 Database Scripts and related Cronjobs

Go to the MI_Scripts directory you downloaded from the SVN repository to your local drive. Go to MI_Scripts/DBVM. Copy the whole MonitoringInfrastructure directory and paste the contents into /opt/optimis in DBVM (create first /opt/optimis if not present).

In DBVM, go to /opt/optimis/MonitoringInfrastructure. Then, go to the share directory and edit database.properties. Enter the MI database connection information, save the file and close it. Go back to /opt/optimis/MonitoringInfrastructure. Then, go to the cron directory. Open the file named "cronfile.database" and create a cronjob in DBVM according to the information provided in cronfile.database.

Location of installation: DBVM.

2.2.4 Nagios Server

The Nagios server is required on one physical host in order to monitor all the physical hosts of the OPTIMIS cloud. The installation of the Nagios server on Linux can be usually done via repositories. Example on CentOS:

```
yum install nagios
```

Also, Nagios plugins are required on that same physical host:

```
yum install nagios-plugins
```

Next step: install httpd. Example on CentOS:

```
yum install httpd
```

Make sure that Nagios and httpd are automatically started at boot time by executing the following commands:

```
chkconfig nagios on
```



chkconfig httpd on

Location of installation: one physical host belonging to the OPTIMIS cloud and taking the role of Nagios server.

Nagios server version: 3.2.3 or higher.

2.2.4.1 NRPE Server & Plugins

The NRPE (Nagios Remote Plugin Executor) daemon and Nagios plugins are required on every physical host of the OPTIMIS cloud that is monitored by the Nagios server. The installation of NRPE on Linux can be usually done via repositories. Example on CentOS:

```
yum install nagios-nrpe
yum install nagios-plugins
```

Location of installation: every physical host belonging to the OPTIMIS cloud that must be monitored by the Nagios server.

Nagios NRPE server version: N/A.

2.2.5 Nagios Configuration and Plugins on Nagios Server

2.2.5.1 Cfg Files

On the Nagios server, go to <Nagios_dir>/etc (for instance, /usr/local/nagios/etc). Make a copy of the objects directory (to keep a safe copy):

```
cp -rf objects objects.sav
```

Go to the MI_Scripts directory you downloaded from the SVN repository to your local drive. Go to MI_Scripts/Physical_Hosts/Nagios/etc. Copy the whole objects directory and paste the contents into <Nagios_dir>/etc on the Nagios server. Now, you are going to have to adapt the config files for your own OPTIMIS cloud. Open and edit the following files:

- In etc:
 - o nagios.cfg:
 - Enable the following object config files: commands.cfg, contacts.cfg, timeperiods.cfg, templates.cfg, hosts.cfg, hostgroups.cfg, services.cfg, localhost.cfg
 - Set status_update_interval=10
 - Set check_external_commands=1
 - Set command_check_interval=-1



- In etc/objects:
 - o contacts.cfg: change the email address of the Nagios admin user.
 - o hostgroups.cfg: define the groups of hosts that must be monitored. The host names listed in this file must refer to the hosts listed in hosts.cfg.
 - o hosts.cfg: this file lists all the remote physical hosts that can be potentially monitored by means of NRPE. You can define OPTIMIS hosts, i.e. physical hosts belonging to your cloud. You can also define non-OPTIMIS hosts, i.e. physical hosts located elsewhere. This distinction is actually artificial, since in both cases they will be monitored via NRPE. Warning: only the remote hosts grouped in hostgroups.cfg and referenced in hosts.cfg will be actually monitored.
 - o localhost.cfg: change the IP address to the one of your Nagios server.

Location of installation: physical host selected as Nagios server.

2.2.5.2 Libexec Files

On the Nagios server, go to <Nagios_dir> (for instance, /usr/local/nagios). Make a copy of the libexec directory (to keep a safe copy):

```
cp -rf libexec libexec.sav
```

Go to the MI_Scripts directory you downloaded from the SVN repository to your local drive. Go to MI_Scripts/Physical_Hosts/Nagios. Copy the whole libexec directory and paste the contents into <Nagios_dir> on the Nagios server. Make sure all the files of libexec are executable.

Also, you must install the mcelog utility to monitor possible hardware failures on the Nagios server. That utility is called by the check_optimis_hardware_error.sh script (see next table below). To install mcelog on CentOS:

```
yum install mcelog
```

After the installation, test mcelog by running:

```
/usr/sbin/mcelog --ignorenodev --filter >> /var/log/mcelog
```

Then, add this line to the crontab:

```
0 * * * * /usr/sbin/mcelog --ignorenodev --filter >> /var/log/mcelog
```

Note: this is the “old school” way of running mcelog hourly. It is also possible to set up mcelog as a daemon running continuously (cf. <http://www.mcelog.org/installation.html>).



Below is the list of scripts from libexec (with their arguments when applicable) that are used by the Monitoring Infrastructure (which means that they are listed in the Nagios config files in order to be called every x seconds by Nagios):

check_users -w 5 -c 10
check_disk -w 20% -c 10% -p /dev/hda1
check_procs -w 5 -c 10 -s Z
check_procs -w 150 -c 200
check_load -w 15,10,5 -c 30,25,20
check_disk -w 20% -c 10% -p /
totalmem.sh
freemem.sh
check_optimis_users.sh
check_optimis_mac_address.sh
check_optimis_fqdn.sh
noproc.sh
check_downstream.sh -i eth0
check_upstream.sh -i eth0
check_optimis_reboot.sh
check_status -H localhost -w 3000.0,80% -c 5000.0,100% -p 5
check_optimis_disk_total_space.sh
check_optimis_cpu_speed.sh
check_optimis_hardware_error.sh

Go to the libexec directory and test each and every one of these scripts manually (example: `./check_load -w 15,10,5 -c 30,25,20`). Check if their standard output is OK. Some of them may need adaptation depending on your Linux OS.

Location of installation: physical host selected as Nagios server.

2.2.5.3 Nagiosity

Go to the MI_Scripts directory you downloaded from the SVN repository to your local drive. Go to `MI_Scripts/Physical_Hosts/Nagios/`. Copy the whole nagiosity directory and paste the contents into `<Nagios_dir>` on the Nagios server.

On the Nagios server, go to `<Nagios_dir>/nagiosity` (for instance, `/usr/local/nagios/nagiosity`). Then, go to `configobj-4.7.2` and run:

```
python setup.py
```



Then, go back to <Nagios_dir>/nagiosity and edit the property file named "status2xml.properties". Make sure the path going to status.dat is valid. Correct it if necessary. Status.dat can be usually found in <Nagios_dir>/var.

Location of installation: physical host selected as Nagios server.

2.2.5.4 Apache Config File

On the Nagios server, go to /etc/httpd/conf.d. Edit nagios.conf. Add the following lines to the config file:

```
ScriptAlias /nagios/xml "/usr/local/nagios/nagiosity"  
<Directory "/usr/local/nagios/nagiosity">  
    Options ExecCGI  
    AllowOverride None  
    Order allow,deny  
    Allow from all  
</Directory>
```

Warning: correct the value of the Nagios path in case the Nagios server was not installed in /usr/local.

Location of installation: physical host selected as Nagios server.

2.2.6 Plugins and NRPE Configuration on monitored physical Hosts

2.2.6.1 Libexec Files

On every remote physical host monitored by the Nagios server, go to <Nagios_dir> (for instance, /usr/local/nagios). Make a copy of the libexec directory (to keep a safe copy):

```
cp -rf libexec libexec.sav
```

Go to the MI_Scripts directory you downloaded from the SVN repository to your local drive. Go to MI_Scripts/Physical_Hosts/Nagios. Copy the whole libexec directory and paste the contents into <Nagios_dir> on every remote physical host monitored by the Nagios server. Make sure all the files of libexec are executable.



Also, you must install the mcelog utility to monitor possible hardware failures on every remote physical host monitored by the Nagios server. That utility is called by the `check_optimis_hardware_error.sh` script (see next table below). To install mcelog on CentOS:

```
yum install mcelog
```

After the installation, test mcelog by running:

```
/usr/sbin/mcelog --ignorenodev --filter >> /var/log/mcelog
```

Then, add this line to the crontab:

```
0 * * * * /usr/sbin/mcelog --ignorenodev --filter >> /var/log/mcelog
```

Note: this is the “old school” way of running mcelog hourly. It is also possible to set up mcelog as a daemon running continuously (cf. <http://www.mcelog.org/installation.html>).

Below is the list of scripts from libexec (with their arguments when applicable) that are used by the Monitoring Infrastructure (which means that they are listed in the Nagios config files in order to be called every x seconds by Nagios):

<code>check_users -w 5 -c 10</code>
<code>check_disk -w 20% -c 10% -p /dev/hda1</code>
<code>check_procs -w 5 -c 10 -s Z</code>
<code>check_procs -w 150 -c 200</code>
<code>check_load -w 15,10,5 -c 30,25,20</code>
<code>check_disk -w 20% -c 10% -p /</code>
<code>totalmem.sh</code>
<code>freemem.sh</code>
<code>check_optimis_users.sh</code>
<code>check_optimis_mac_address.sh</code>
<code>check_optimis_fqdn.sh</code>
<code>noproc.sh</code>
<code>check_downstream.sh -i eth0</code>
<code>check_upstream.sh -i eth0</code>
<code>check_optimis_reboot.sh</code>
<code>check_status -H localhost -w 3000.0,80% -c 5000.0,100% -p 5</code>
<code>check_optimis_disk_total_space.sh</code>
<code>check_optimis_cpu_speed.sh</code>
<code>check_optimis_hardware_error.sh</code>



Go to the libexec directory and test each and every one of these scripts manually (example: `./check_load -w 15,10,5 -c 30,25,20`). Check if their standard output is OK. Some of them may need adaptation depending on your Linux OS.

Location of installation: every remote physical host monitored by the Nagios server.

2.2.6.2 NRPE Config File

Go to the MI_Scripts directory you downloaded from the SVN repository to your local drive. Go to MI_Scripts/Physical_Hosts/Nagios/etc_of_hosts_monitored_with_NRPE. Copy nrpe.cfg and paste it into `<Nagios_dir>/etc` of every remote physical host monitored by the Nagios server.

Open nrpe.cfg. Go to the bottom of the file, to the section entitled "COMMAND DEFINITIONS". Check that the paths going to the Nagios directory and used for the commands are valid. Usually, the Nagios directory is `/usr/local/nagios`. Correct the path values if needed.

Location of installation: every remote physical host monitored by the Nagios server.

2.2.7 Apache Tomcat

The Apache Tomcat container is required. This is a pre-requisite for an installation step described further down in this document that relates to the installation of three WAR files.

If Tomcat is not yet installed in IPVM, install it. The installation of Tomcat on Linux can be usually done via repositories. Example on CentOS:

```
yum install tomcat6
```

Location of installation: IPVM.

Apache Tomcat version: 6.0.32 or higher.

2.2.8 Java RESTful Web Services

Check that the following WAR files have been deployed in the Apache Tomcat container of IPVM.

- MonitoringManager.war
- Aggregator.war
- MonitoringManagerWeb.war

If they are not present in the container, get them from the SVN repository and deploy them.

The next step deals with the configuration of a property file. In IPVM, go to `/opt/optimis/etc/MonitoringManager` and open the file named "mmanager.properties". Enter the MI database connection information, save the file and close it.

Location of installation: IPVM.

2.2.9 MySQL Client

The MySQL client is required in the IPVM. The installation of the MySQL client on Linux can be usually done via repositories. Example on CentOS:

```
yum install mysql
```

Location of installation: IPVM.

MySQL client version: N/A.

2.2.10 RabbitMQ

Go to the MI_Scripts directory you downloaded from the SVN repository to your local drive. Go to `MI_Scripts/IPVM`. Copy the whole `MonitoringInfrastructure` directory and paste the contents into `/opt/optimis` in IPVM (create first `/opt/optimis` if not present).

In IPVM, go to `/opt/optimis/MonitoringInfrastructure/rabbitmq`. Open README and follow the instructions provided in that file.

Location of installation: IPVM.

2.2.11 Collector Scripts and related Cronjobs

In IPVM, go to `/opt/optimis/MonitoringInfrastructure`. Open README and follow the instructions provided in that file.

Location of installation: IPVM.

2.2.12 OpenNebula Monitoring Plugin

Instructions on how to install the OpenNebula plugin for running VMs:

- go to the OpenNebula directory in the optimis SVN:
`optimis/branches/OptimisY3/MonitoringInfrastructure/Collectors/OpenNebulaVirtual`



- update the `src/main/resources/config.properties` file with regards to `onl.userpd` and `onl.host.oca` parameters.
- compile and generate war file:
`mvn -PTST package cargo:redploy`
- copy the `target/OpenNebulaVirtual.war` file into the IPVM, specifically the Tomcat's `webapps` directory or deploy the war through Tomcat Manager interface.
- update the `virtual_collector.properties` file in the IPVM (location: `/opt/optimis/MonitoringInfrastructure/scripts`) or in the below SVN location:
`optimis/branches/OptimisY3/MonitoringInfrastructure/MI_Scripts/IPVM/MonitoringInfrastructure/scripts`
and set the below parameter to OpenNebula:
`CLOUDMIDDLEWARE=OpenNebula`
- test the plugin by typing:
`curl http://localhost:8080/OpenNebulaVirtual/virtual/data/str`

Location of installation: IPVM.

2.2.13 OpenStack Monitoring Plugin

Instructions on how to install and use the OpenStack's monitoring plugin for running VMs:

- go to the OpenStack directory in the optimis SVN:
`optimis/branches/OptimisY3/MonitoringInfrastructure/Collectors/OpenStack`
- change the permission of the script to be executable (if not previously done):
`chmod 755 vm_collector.sh`
- copy the `vm_collector.sh` file into the physical nodes. The preferred location is `/opt/optimis/MonitoringInfrastructure`
- make sure the **virt-top** package is installed in the physical nodes, and check the output. The "Domain ID" column should be in position 19. However, it may also be in the 18th position.
- modify the `virtual_collector.sh` script in the IPVM (location: `/opt/optimis/MonitoringInfrastructure/scripts`) or in the below SVN location:
`optimis/branches/OptimisY3/MonitoringInfrastructure/MI_Scripts/IPVM/MonitoringInfrastructure/scripts` for the following parameters:

```
# list of nodes
```

```
LIST="bscgrid21 bscgrid22"
```

```
# location of storing the vm_collector.sh script in the node
```

```
DIR=/opt/optimis/MonitoringInfrastructure
```

```
# public key to access the node
```

```
KEY=root_OPTIMIS
```

- update the `virtual_collector.properties` file in the IPVM (location: `/opt/optimis/MonitoringInfrastructure/scripts`) or in the below SVN location:



optimis/branches/OptimisY3/MonitoringInfrastructure/MI_Scripts/IPVM/MonitoringInfrastructure/scripts and set the below parameter to OpenStack:
CLOUDMIDDLEWARE=OpenStack

- Make sure Aggregator has been deployed in Tomcat. Then, test the plugin by typing:
./virtual_collector.sh
The output would be an XML string.

Location of installation: IPVM for *virtual_collector.sh* and physical hosts for *vm_collector.sh*.

2.2.14 Service Level Monitoring and Probes

Application level data from inside the service, such as the number of requests/second on a web server or the number of jobs in queue for a calculation service, can be monitored and used for statistics and elasticity. This is a two-step process: data is sent from inside the service to a gateway collector component which in turn inserts the data into the monitoring system.

Service MonitoringCollector Component

Requires Java JRE 6.0 or later.

The collector component (in MonitoringInfrastructure/Collectors/ServiceMonitoring) is stand-alone and can either be run straight from the Optimis source tree:

```
mvnexec:java
```

or compiled into an independent .jar file:

```
mvnassembly:single
```

(copy the *-with-dependencies.jar file to the host where it should run)

```
java -jar *-with-dependencies.jar
```

It is recommended to run the component using the nohup command (or equivalent) or inside a screen to ensure it continues to run after the user logs out.

Configuration

The collector reads configuration data from a file called *monitoring.properties* that can either

1. Be specified using the -Dconfiguration.file.path= setting of the JVM,
2. be located in the directory /etc/optimis/modules/service-monitoring/,
3. be located in the current working directory.

If the property file is present in more than one location, the order of precedence is as listed above (e.g. -Dconfiguration.file.path will overrule the other two).

The settings that can be specified are:

Field	Semantic
monitoring_host	Endpoint of the Monitoring System
monitoring_port	Port of the Monitoring System
cloudoptimizer_host	Endpoint of the Cloud Optimizer



cloudoptimizer_port	Port of the Cloud Optimizer
own_port	Local port that the collector should listen to

Probes

Probes are small applications or code snippets that run inside each service, collect the data to be monitored, and send the data to the service monitoring collector component. The probes can be implemented in any language, the only requirement is that the resulting data must be sent to the collector component using REST.

The format of the data has to be as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<measurement>
<data>52</data>
<instanceID>workernode_12</instanceID>
<name>queuelength</name>
<serviceID>jobservice</serviceID>
<timestamp>1362124650207</timestamp>
</measurement>
```

The fields have the following semantics:

Field	Semantic
data	Measurement of KPI
instanceID	ID of executing instance
name	KPI name
serviceID	ID of service
timestamp	Creation time of measurement

It is important to note that each service will require unique probes for its specific KPIs, environment and language, and hence no “standard” probe exists. However, the Optimis package includes two sample probes:

- processprobe: uses python and the linux ‘top’ command to count the number of running “httpd” processes.
- JMXprobe: implemented using Java, this probe uses JMX to query a Java JVM for information exposed through JMX beans. Any property exposed as such bean can be used. Requires Java JDK 6.0 or later.

Feel free to modify these sample probes to suit your specific service. POST the resulting data to the collector component (the IP and port are printed when the component is started). Note that the data should be posted to `http://<address:port>/data`, where address and port refer to where the collector is running.