Name: Matthias Rechenburg
Email: matteverywhere@googlemail.com
Organization: the openQRM project

Automated system and service monitoring with openQRM and Nagios

The first step to make sure all systems and services in a data centers are running well is to monitor them. A well-known, proven and widely used monitoring tool is Nagios which is available for openQRM in the flavor of an additional plugin. The second, also essential, step is the automatic handling of errors, what openQRM is famous for.
The combination of the enhanced monitoring utility Nagios and the automated error-handling, high-availability and fail-over features of the openQRM data center management platform creates a powerful and dynamic environment which reduces down-time of systems and services in a modern data center to the minimum.

Common data center scenarios and their challenges

In a common data center setup there are different server "islands" for specific purposes as databases, web-servers, infrastructure and communication systems, network-devices, storage-servers, etc. Also, often there is a separation between production-, QA- and development environment. Each section of the data center has its own specific setup, mostly a very special, static installation, maintained by one or more administrators. To make it worse, each section may have different service-level-agreements (SLA) which generates additional complexity for the service- and system-monitoring infrastructure. Also common for data centers is that new systems are added sequentially while other systems are fully underutilized, just consuming power and producing heat.

Turning to fully automated system- and service monitoring

With the increasing number of systems and the resulting complexity the goal is to reach a level of fully automated service- and system-monitoring which is easy to setup and maintain and which provides an automated error-handling to gain high-availability at system- and service-level for the complete data center environment.

Nagios in General

Nagios is a widely adopted open-source host, service and network monitoring program based on a client/server concept. The Nagios-server gets monitoring informations by active- or passive-checks. That means it either actively tests the availability of a system or service from the Nagios-server itself or it passively receives informations about tests running on the remote system. The passive checks are initiated by the Nagios-client (nrpe) running on the systems monitored by the Nagios-server. The client part of Nagios is designed in a plug-able way. New service checks can be added easily by creating new Nagios-plugins interfacing with the plugin-architecture of the base Nagios-server.

The Nagios-server runs within an Apache web-server and consists of perl- and shell scripts mixed with binary tools executed via the CGI-interface.
In configurable intervals it checks various services like SMTP, POP3, HTTP, NNTP, PING, etc. and provides the gathered data center informations in a nice web-interface. It also monitors system resources like CPU-load, memory and disk usage, running processes, logs, etc. and environmental factors such as CPU-temperature.

openQRM in General

openQRM is an open source system management platform which integrates existing components to manage enterprise data centers efficiently. The out-of-the-box functions include, but are not limited to, provisioning, monitoring, reporting and high availability. Other functionality and integration with third-party software like VMware, Xen, Qemu and Linux-VServer as partitioning technologies, Ldap-Authentication, Iscsi (booting from Iscsi-targets), support for different operation systems and more are available.
The openQRM-server itself is based on a Tomcat-httpd server with an underlaying Mysql database. Its basic services are implemented in Java, the graphical User-interface is written in JSP and its actions are executed by shell-scripts.

openQRM is designed with a plug-able architecture which allows developers to add features and components without changing any code in the base server. Each plugin is self-contained and can add functionality to the openQRM engine or change existing mechanisms via plugin extensions. Those plugin extensions are setup via a xml configuration file which hooks into the extensions points provided by the base openQRM-server, e.g. a plugin can provide a ServerService by creating an xml file which includes the configuration for the service. The ServerService extension takes this configuration and runs the service according to its configuration. In the same way a plugin can provide additional web-pages, create new menus, listen to events in the openQRM-server and react to them, setup ResourceServices which are then started on the managed servers etc. Also a plugin can interact with the data center objects of the main openQRM-server directly by providing additional jar-packages which are plugged into openQRM automatically.

openQRM manages servers and their services by adding a logical layer called “Virtual-Environments” (VE's). Those VE's are user-defined combinations of “boot-images” and “file system-images” including deployment configuration. A boot-image consist of a Linux-kernel, a special initrd and the kernel-modules. The file system-images are the actual root-file systems of the servers including their applications and services. According to the Virtual-Environment provisioning configuration the openQRM-server then selects the best fitting hardware (or virtual-partition) to start. This abstraction eases-up server administration, management and deployment efficiently.

Automated installation of Nagios within openQRM

The Nagios-plugin provides an automated installation of the Nagios-server and client within the openQRM environment. It requires to have the openQRM Apache-plugin to be installed before to stage the web-server component needed by Nagios. By enabling the Nagios-plugin in the openQRM-server the Nagios-server automatically gets initialized and started. Its web-interface fits into the openQRM-user-interface and is accessible from the main “Management Tools” menu. During the installation of the Nagios-plugin it sets up a ResourceService (a plugin-extension supplied by the base openQRM-server) which dynamically preconfigures the Nagios-clients according to their VE-profile and start them on the managed systems automatically.

Binary packages for the Nagios- and Apache-plugin are provided for various different Linux distributions such as : CentOs, Fedora, Suse and Debian. Thanks to the automated build-system within openQRM the Nagios- and Apache-plugin also can be built from the sources easily.

Contents of the Nagios-plugin

The Nagios-plugin provides several components which are interfacing with the plugin-architecture of the openQRM-server. In detail it adds the following services to the base server.

- Nagios server start/stop via a ServerService extension
- Nagios menu integration via menu extensions
- Nagios server-pages integration via Web-page extensions
- Nagios client start/stop on the managed servers via a ResourceService extension
- Alert transfer from Nagios to openQRM via Nagios-check
- Nagios service-check configuration via a Web-page extension
- CPU, memory and network graphs by rddtool managed via a ResourceService extension
- Automated creation of Nagios configuration files for the VE's via the EventListener extension

Separately, all components are setup in the plugins xml config file and automatically activated when the plugin gets enabled in openQRM.

Automated VE configuration

The integration of Nagios within openQRM provides an automated setup and configuration of Nagios for the managed servers. On an additional Nagios-service configuration page the system-administrator can select the services to be checked per system, easily. When submitting a new monitoring setup for a VE a Nagios-configuration file is created. The Nagios-client ResourceService on the managed servers downloads this configuration file during boot-up and starts the Nagios-client according to the VE-profile configuration. This ResourceService supports having multiple Nagios-client packages for different Linux-distributions, which is especially relevant for a data center environment consisting of servers running a mixture of different Linux distributions and versions.

As soon as the Nagios-client has been started it reports the informations gathered from the passive checks (service/system checks running on the remote system) back to the main Nagios-server running embedded within the openQRM-server.

Cpu, memory and network graphs

Additional to the monitoring and automated error-handling mechanisms the Nagios-plugin for openQRM provides real-time graphs and history reports about network-traffic, CPU-utilization and memory usage of the Virtual-environments. Especially the history reports are useful to compare e.g. system uptime and service availability with the service-level-agreements for a specific system. To gather the system informations on the managed servers the “sar” utility is used, which is reporting the monitoring data to “rddtool”. This then creates the servers statistic in a graph embedded into the openQRM-user-interface.

High-availability for the managed servers

openQRM takes care about high-availability on application and system level by either initiating fail-over of an application to a "hot-standby" or fail-over of full servers including their applications to available resources using openQRM's rapid-deployment features. For application level high-availability a Virtual-Environment is configured in active-active mode so that an application in error-state can be quickly re-covered on a “hot-standby” system. On system-level openQRM re-deploys a crashed server automatically on another, available resource in the data-center when AHR (automatic hardware recovery) is enabled for the VE. To give an overview of the high-availability status of the complete data center and resources, available and selected for possible system- or service fail-over openQRM calculates an high-availability pool according the HA-configuration and requirements of the active VE's. System and application high-availability for the managed servers can be combined to gain a “unbreakable service”.

Automated error-handling

To limit possible human errors during manual fail-over situations Nagios monitoring alerts are transferred to the openQRM-server as events.

These are then evaluated and handled by, e.g. initiating a system- and/or service fail-over to one or more resources available in the high-availability pool. The transfer of the Nagios alerts to openQRM events is provided as an additional Nagios-service command.

Directly, this mechanism connects critical services in the data center through Nagios to the error handling procedure on the openQRM server.
It also avoids the need to look up several monitoring utilities for different server environments. because all data-center related events are collected in the single management console of openQRM.
For the automated error-handling in case of an system- or service fail-over caused by an alert openQRM selects the resource which fits best for a potential fail-over according the VE's profile. In the VE-profile the special requirements for a specific service are defined, like special hardware, amount of CPU- and memory needed , physical network cards, network-configuration and more. When using automatic deployment the VE-profile assists openQRM to find the systems which are fitting best for the VE-service.
Additional resources in openQRM can be tagged so that the mechanisms which select the resource for fail-over only choose servers marked with a special VE-tag.

As an example of the automated error-handling facilities, a web-server Virtual-environment is additional monitored via the Nagios httpd-check by enabling the httpd-check in the Nagios-check configuration page within the VE-configuration. As soon as the web-server Virtual-environment gets activated the Nagios-server, running as a plugin ServerService on the openQRM-server system, sequentially monitors the httpd-service on the target system.
When Nagios detects an service failure or downtime of the httpd-server, either because the httpd-service crashed or the system on which the httpd-service was running is not available any more, it sends an error event on behalf of the resource belonging to the VE to the openQRM-server. In case AHR (automatic hardware recovery) is enabled for the VE the openQRM-server takes care to really shutdown the eventual unavailable system automatically by the help of the openQRM power-management plugins and rapidly re-deploys the web-server Virtual-Environment to another, available system from the high-availability pool.
Also additional actions can be connected to the error-events sent by Nagios, e.g. a web-server VE can also be configured as an active-active cluster with one or more hot-standbys. In case of a service failure, reported by Nagios, the resulting event in openQRM can trigger an application fail-over to its hot-standby within the high-availability cluster.
Combining both methods of high-availability (HA on application and system-level) results in a nearly unbreakable service which always fits to its service-level-agreements and which can be maintained easily.

Distributed Server Configuration

The Nagios-plugin for openQRM supports a distributed Nagios-server environment which is very useful for big data-centers and huge server-farms. The Nagios-server, embedded via the openQRM-plugin, acts as the main console for all other Nagios-servers in the data-center, controls them and receives alerts from them. The distributed setup helps to gain global system- and service monitoring and automated error-handling for the complete IT infra-structure by consolidating the administration to a single management console.

Reference installation

A typical installation of openQRM in an enterprise data-center is configured in an active-passive high available setup using one or more hot-standby systems for the openQRM-server. This prevents it from being the single point of failure, as it is the central point of management. Additionally, the openQRM-server can be setup in a distributed manner to gain performance and reduce possible down-times by using high available components like a HA-storage-server and a remote HA-database. The Nagios-plugin is installed to provide enhanced system and service monitoring and to connect alerts directly to the automated error-handling procedures in the openQRM-server. Supplementary, dedicated storage-server featuring advanced storage-techniques like snapshotting and cloning are used to host the filesystem-images. This standard openQRM environment is enhanced via additional plugins for Ip- and Vlan-management (network and network-provision plugins), storage integration (Lvm-mgmt and Netapp-plugin), virtualization technologies as needed (VMware, Xen, Qemu, Linux-VServer) and configuration and service management on the managed servers via the Puppet- and Webmin-plugin.

Summary and Conclusion

The open architecture of openQRM, its unique features paired with lots of automatism, and flexibility in data-center management results in better scalability, better performance, faster deployment and it decreases services down-times and eases up the maintenance of the infrastructure. openQRM's generic approach to integrate and implement existing system management components into a single, unified management console drives IT-management away from dependencies by vendors, specialists or special “in-house” created tools through a certified, standardized and open data-center management platform. The integration with Nagios expands openQRM's system and service monitoring facilities and importantly supplies automated error-handling by linking Nagios error-events to error-handling procedures within openQRM. openQRM's unique high-availability features, its powerful system management capabilities combined with the enhanced monitoring capabilities of Nagios plus lots of other plug-able features like managing different types of virtualization, automated Vlan- and Ip-Administration, LDAP-authentication, fully automatic deployment and multi-architecture support makes this platform an essential tool in modern data-centers.

Related Urls

Nagios Home page : http://www.nagios.org
openQRM Home page: http://www.openqrm.org
openQRM Project page: http://sourceforge.net/projects/openqrm
openQRM community Wiki : http://linuxfellaz.net/