Conference proceedings
5th TYPO3 Conference

Frankfurt, Germany
10th – 12th october 2009

TYPO3 website monitoring with Caretaker

Martin Ficzel, Thomas Hempel, Christopher Hlubek, Tobias Liebig

Introduction

As websites and web-applications are crucial parts of many businesses it is important to manage the security and availability of such systems. TYPO3 websites are complex systems consisting of many different parts from many different parties (core system, extension authors, custom code). A holistic monitoring approach is hard to achieve with existing solutions. Many years of TYPO3 administration, hosting and maintenance showed the need of a specialised solution to lower the maintenance complexity and to increase the reliability and security.

1 How to monitor a TYPO3 Website?

1.1 Using an infrastructure monitoring solution (e.g. *Nagios*)

Bad idea

- complex setup for unexperienced users
- restricted by the available checks (e.g. only ping and HTTP)
- a lot of integration scripts / plugins have to be written by the administrator
- management and maintenance is complicated for the many aspects of monitoring needed for TYPO3 (especially with changing security rules)
- even simple checks like "Specific extension version" are complicated to implement outside of the TYPO3 system (TYPO3 has no remote services to query for specific information)
- client installations not fully under control are hard to integrate if the monitoring relies on custom scripts / plugins (e.g. Bash scripts or NRPE plugins)

Good for

- general checks of server hard- and software (operating system, Apache, MySQL, . . .)
- notifications / escalation strategies
- central IT infrastructure monitoring

1.2 Using a specialized monitoring solution for TYPO3

The idea

Having a central monitoring system that can monitor many TYPO3 installations wherever they are located and making the management and maintenance as easy as possible. The monitored TYPO3 systems are equipped with a small sensor extension that can carry out basic operations and answer questions about the system (like installed extensions, TYPO3 version, backend users, etc.) in a secure way.

Requirements

- secure communication
- monitoring rules and logic live on the central server
- the sensor extension should be as simple as possible
- the system should be easily extensible
- the logic structure of the websites (customers, servers, different TYPO3 versions) should be representable
- customers should have access to monitoring information about their websites
- "plug-and-play" for TYPO3 monitoring

2 The Caretaker TYPO3 extension

Based on these requirements n@work Hamburg and networkteam Kiel / Hamburg joined forces to develop a solution named *Caretaker*. The solution consists of multiple TYPO3 extensions for the central server and the remote systems (instances).

2.1 Definitions

The system defines some aspects for its configuration and management. To better understand the solution and its flexibility the following definitions are helpful.

Caretaker server The central monitoring server based on TYPO3

Caretaker instance A TYPO3 installation to be monitored equipped with the sensor extension (`caretaker_instance`)

Instance group A group of Caretaker instances or other instance groups

Test service A configurable check on the Caretaker server (e.g. "Check for extension version", "Check TYPO3 version")

Test A configured test service on the Caretaker server

Test group A group of tests or other test groups

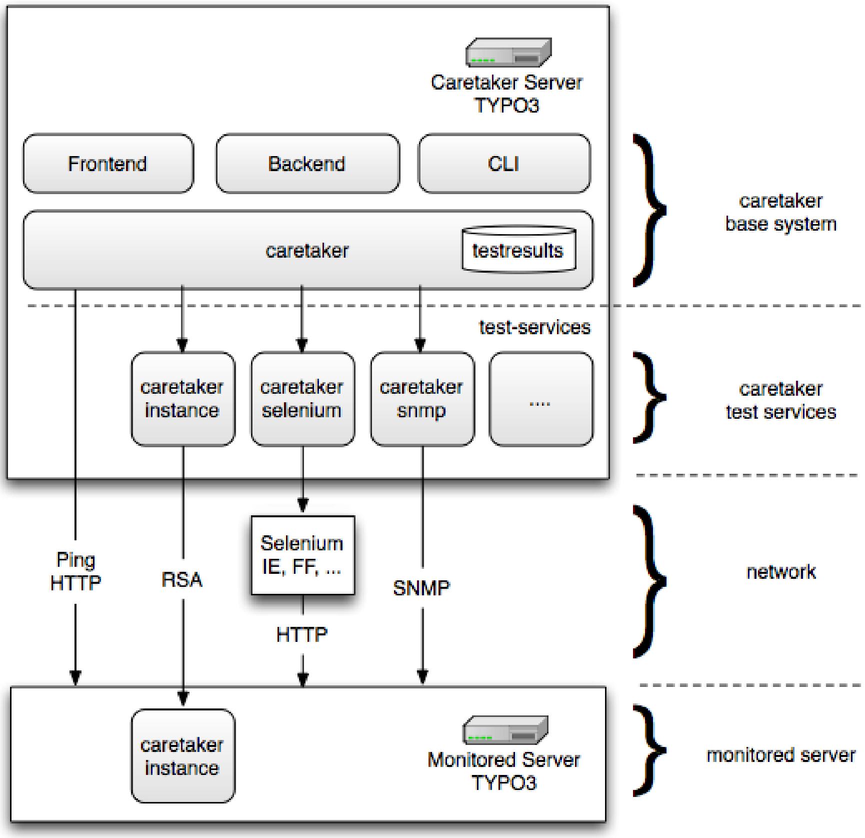
Node A node can be an instance or instance group as well as a specific test or test group of an instance

Remote operation An operation that can be called on an instance by the Caretaker server and returns information

2.2 Configuration structure

The flexible configuration allows for a good representation of the logical structure of websites of different customers, on different servers, with different TYPO3 versions but shared tests. This helps to keep the configuration DRY and eases the management and maintenance. The configuration structure is suitable for small installations as well as large scale monitoring of heterogeneous installations.

2.3 System architecture



Caretaker server

The caretaker server is responsible for the central management, execution of tests, evaluation of the results, the storage of aggregated test results and reporting as well as notifications. The server runs on a dedicated TYPO3 installation and is managed through the TYPO3 backend.

Caretaker instance

Any TYPO3 installation can be an instance if the `caretaker_instance` extension is installed. Every instance has a public / private key pair for securing and authenticating the access from the Caretaker server. A special web service exposes the remote operations and is the single entry point for the server access.
Remote operations are predefined by a set of classes that implement a simple interface. The default operations are read-only and don't expose security critical information.

Test services

Test services have common configuration options for test execution intervals, notifications. Tests can be grouped in test groups which can be assigned to instances or instance groups, which allows for a reuse of common tests. Tests are always executed on the Caretaker server but can carry out remote operations on instances. The logic for specific checks like "Are unsecure extensions installed" and evaluation of operation results is contained in the test services.

Test execution and monitoring can be done through a backend module. For automatic test execution a CLI interface is provided, which can be used to selectively execute tests by hierarchy.

Backend

The system configuration and test status is displayed as a tree and can be configured in the Caretaker overview module. For each hierarchy level the system status and an overview of failed / successful tests is shown. Single tests can be re-executed on demand (e.g. when a problem is fixed).

Frontend

A set of plugins is provided to display information about nodes. This can be used to build a frontend for customers or a simplified technical overview.

2.4 Use cases

Already implemented

- HTTP test with basic performance monitoring (by request time)
- Ping test (with thresholds for latency)
- SNMP (Simple Network Management Protocol) tests
- NRPE (Nagios Remote Plugin Executor) tests to call nagios remote plugins on hosts
- TYPO3 version test (minimum, maxium version)
- Extension version and status test (required, forbidden)
- Unsecure extension test (marked as unsecure by the security team)
- Backend user test (required, forbidden)
- Install tool enabled test

Planned

- Password quality test
- Selenium tests for integration tests
- Core modification / extension modification test (with MD5 checksums)
- System status test from report module (in 4.3)
- Record modification time test

2.5 Extensibility

The caretaker server and instance can be easily extended with new test services and remote operations. The open architecture allows for the monitoring of arbitrary services.

Conclusion and outlook

The caretaker solution is already in production for some pilot projects and proved itself valuable. The technical architecture is a solid foundation for future development. The open source character of the solution should motivate future users to integrate their experiences and knowledge into the project.