Qosium Storage

Qosium Storage is the results system of Qosium. It comprises a QMCP receiver, a dedicated database server, and a web user interface for visualization and accessing the results. Results can also be accessed by using REST API. Storage is handy when results data grows large, and data access is wanted to be centralized. It is a key component in monitoring setups.

Table of Contents

1. Introduction

1.1. Basics

When Qosium is used for monitoring, statistics are generated continuously. Often, the monitoring setup consists of multiple simultaneous measurements, which can create a lot of monitoring statistics. Handling this by using separate results files becomes clumsy and, more importantly, lacks a way to give the user a complete picture of the status of the monitored system.

Qosium Storage is the answer to this need, providing permanent storage to the results and easy centralized access to them. Situation awareness rises to a new level as the QoS status of the monitored system can be seen as a whole.

Besides monitoring scenarios, Qosium Storage can also become handy in large-scale measurement setups.

In addition to Kaitotek's Qosium Storage, there is a possibility to use your own results system. If you are interested in this, check the QMCP integration possibilities.

The results stored in Qosium Storage can be accessed in various ways:

- QoS heatmap
- Customized overview reports
- List of measurements => raw results
- REST calls for precalculated overview statistics
- REST calls for raw results
- Direct database access

1.2. Structure

Qosium Storage is composed of the following main components:

- QMCP results receiver,
- Database, and
- Web server.

QMCP Receiver is a server listening to Qosium measurement results to be received in the defined port. The web server plays an essential role since it provides the web user interface to Qosium Storage. In addition, Storage supports direct REST calls for querying the results for integration with current applications and services. The results are stored in a database, also enabling direct access to the results with SQL queries.

All the components can be used and turned on and off separately. In the optimal case, they all are installed on the same platform, but it is not mandatory. The components can lie on different devices, making it possible, for example, to use your separate database with Qosium Storage.

1.3. Platforms

The database system of Qosium Storage is, by default PostgreSQL with TimescaleDB. Using other *SQL*-based solutions, however, are also possible. Please reach out to us for more information.

The recommended operating systems for Qosium Storage are **Ubuntu**, **Debian**, and **Redhat/CentOS**. Other operating systems are also possible, so if you have something in mind, ask Kaitotek support about it. In addition, if you have an existing supported database in your current environment, Qosium Storage can be configured to use that.

2. Preparative Work

2.1. Installation

2.1.1. From Installation Package (Debian-based)

In Debian-based systems, Qosium Storage installation is usually done from a deb installation file. The package is typically named `QosiumStorage_<version_details>.deb`, where the `<version_details>` part details depend on the version to be installed.

If you like to see what will be installed and where to, run:

```
dpkg --contents QosiumStorage_<version_details>.deb
```

To install a fresh or upgrade an existing Qosium Storage in a machine, open Terminal and run:

```
sudo dpkg -i  QosiumStorage_<version_details>.deb
```

Alternatively, you can use *apt* for installation:

```
sudo apt install ./QosiumStorage_<version_details>.deb
```

When running the installer, it checks for the dependencies and, if missing them, gives instructions on how to install them:

- Java,
- nginx (please reach out to us if you want to use another web server),
- PostgreSQL, and
- TimescaleDB.

The installation process asks some relevant details while installing:

- Local or remote database
- Web user username
- Web user password
- Database password

- Path for database (local database) or IP address to remote database

You can check which version of Qosium Storage is installed by running:

```
dpkg -l qosiumstorage
```

2.1.2. Manual Procedure

The installation package is unavailable for all platforms, but Qosium Storage will be installed manually. Kaitotek support will help you carry this out step-by-step.

2.2. Parameterization

After being installed, Qosium Storage is ready to be used, and parameterization is hardly needed. In case you do, see the instructions here.

2.3. Running

2.3.1. Control

Qosium Storage will be started automatically after the installation. It runs as a system service, controlled with *systemctl* commands:

Start:

```
sudo systemctl start QosiumStorage
```

Stop:

```
sudo systemctl stop QosiumStorage
```

Request status:

```
sudo systemctl status QosiumStorage
```

Disable Qosium Storage from starting automatically upon reboot:

```
sudo systemctl disable QosiumStorage
```

Enable Qosium Storage to start automatically upon reboot:

```
sudo systemctl enable QosiumStorage
```

2.3.2. License

Qosium Storage installation requires a license to operate. The License will be activated against Kaitotek's license server online. The process is fully automatic and is carried out when Qosium Storage is turned on. All you need to do is ensure that the device where Qosium Storage is installed has Internet access during

activation. Once done, Internet access is no longer required during the active License Period.

2.4. Feeding Storage with Results

2.4.1. The Procedure

Qosium Probes support direct results distribution to external systems such as Qosium Storage. Results sending is activated via a *measurement controller*.

When using Qosium Scope, the instructions to activate results sending are provided here. In the case of Scopemon, see how to enable results distribution. In addition, naturally, also Qosium Scope Lite can be used to activate the feature.

2.4.2. Identifying Qosium Measurements

Before you start pushing measurement results to Qosium Storage, think about how to organize measurement identifications so that it will be fluent in finding particular measurement results in the future. There are many ways to identify Qosium measurements, but three ways rise above others when dealing with Qosium Storage:

- **Service ID**,
- **User ID**, and
- **Measurement Description**.

Service ID is an identification of Qosium Probe that you can parameterize. Thus, it can be used in many ways. Often, it is used as a unique and static identifier of Qosium Probe. When used like this, it also identifies the device where the Probe is located. Further, if the device is, e.g., in a vehicle, and it is the only measurement point in that vehicle, **Service ID** identifies the vehicle. **Service ID** does not have to be unique depending on your use case. You can use it, for example, to identify a group of Qosium Probes. For instance, Probes in a specific network segment could use the same **Service ID** to identify that segment.

User ID is an identifier of the *measurement controller* that you can also set freely. For example, in Qosium Scope, it is set here. It provides another dimension in identification; again, there are various ways to use it. For example, you can use **User ID** to give all measurement controllers a unique ID. Another example is grouping. For instance, all VoIP QoS measurements could be grouped under the same **User ID**, while video QoS measurements could use another **User ID**.

Measurement Description is free text and can contain, besides some special characters, anything you wish. While **Service ID** and **User ID** are typically used as categorical/topological differentiators, **Measurement Description** is often used for details. It could contain, e.g., information on some nuances to separate otherwise similar measurements made with the same equipment. This parameter is defined in the measurement controller. For example, in Qosium Scope it is defined here.

All the presented measurement identifiers are stored with the measurement results. Thus, for a single measurement, you always have **Service ID**, **User ID**, and **Measurement Description**. All these can be used in results searches in Qosium Storage. Qosium provides different ways to identify measurements, and it is up to you how you use them.

3. Glossary

Structured Query Language

A language used in programming and designed for managing and processing data held in a relational database management system.