Collecting data from endpoints

The Kaa logging subsystem is designed to collect records (logs) of pre-configured structure on endpoints, periodically deliver these records from endpoints to Operation servers, and either persist them on the server for further processing or submit to immediate stream analysis. The log structure in Kaa is determined for each application by a configurable log schema. On the Operation server side, there are log appenders which are responsible for writing logs received by the Operations server into the specific storage. A Kaa tenant administrator can define only one log appender per application.

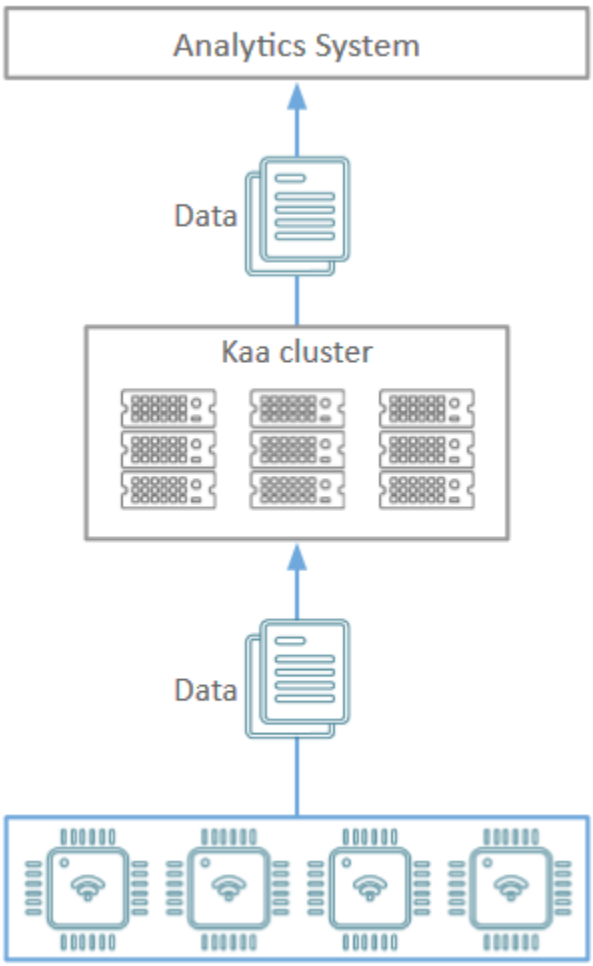
Please view logging system design reference for more background information.

From this guide you will learn how to use the Kaa logging subsystem for data collection.

Basic architecture

The following diagram illustrates basic entities and data flows in scope of the log management:

- The data from the endpoints (logs) is collected and transferred to the server in the format as defined by the log schema created by the developer for the application
- Log appenders submit the logs received by the server to a particular storage or analytics system



Configuring

This section illustrates how to configure a log schema.

Log schema

The default log schema installed for Kaa applications is empty. You can configure your own log schema using the Admin UI or REST API. For the purpose of this guide, we will use schema that is very close to the common log structure: the log level, tag and message.

```
{
    "type":"record",
    "name":"LogData",
    "namespace":"org.kaaproject.kaa.schema.sample.logging",
    "fields":[
        {
            "name":"level",
            "type":{
                "type":"enum",
                "name":"Level",
                "symbols":[
                    "DEBUG",
                    "ERROR",
                    "FATAL",
                    "INFO",
                    "TRACE",
                    "WARN"
                ]
            }
        },
        {
            "name":"tag",
            "type":"string"
        },
        {
            "name":"message",
            "type":"string"
        }
    ]
}
```

Log appenders

Kaa provides default implementations of log appenders that store logs in Hadoop, Cassandra, MongoDB or a local file system (FS). It is possible to develop and integrate custom log appenders.

Coding

Log delivery

The logging subsystem API varies depending on the target SDK platform. However, the general approach is the same.

To transfer logs to the Kaa Operation server, the Kaa client application should use the following code.

Log storage

By default, the Kaa SDK uses an in-memory log storage. Normally, this storage does not persist data when the client is restarted. If this is a concern, Java/Objective-C/C++ SDKs provide a persistent log storage based on SQLite database.

Here is an example how to use SQLite log storage for Java/Objective-C/C++ SDKs and a custom implementation for C SDK:

Log upload strategies

A log upload strategy determines under what conditions Kaa endpoints must send log data to the server. Kaa provides several built-in strategies, namely:

- **Periodic strategy** to upload logs after *at least* the given amount of time passes since the last upload:

NOTE: The decision of whether to upload the logs collected is taken each time a new log record is added. That being said, the next log record added after the time specified passes will trigger a log upload.

- **Log count strategy** to upload logs after the threshold of log records generated is reached:
- **Storage size strategy** to upload logs after the threshold of local log storage space occupied is reached:
- Combined **periodic and log count strategy**;
- Combined **periodic and storage size strategy**;
- Combined **log count and storage size strategy**:

  NOTE: This is the default behavior with log record count threshold of 64 and local storage threshold of 8 KB.

- **Max parallel upload strategy** to limit the number of log batches sent without receiving a response from the server:

Data collection demo

An example application for collecting log data from endpoints can be found in the official Kaa repository.
