hurence

User guide

(C)Hurence

Version v1.0, 18.02.2020: First draft

Table of content

Introduction

This guide will help the user of the data historian in understanding the platform, of its installation, its use. The data historian having been designed with a view to deployment on a Big Data infrastructure, which is by nature an environment complex to implement, we will only cover deployments in this guide on a single machine and gradually. For a deployment on a multi-server Big Data infrastructure and its security, it will be necessary refer to other documentation or contact Hurence at contact@hurence.com. Hurence will typically be able to help you scale your initial single-server infrastructure.

Installation

This section describes a simple installation of data historian on a single server. The Data Historian being designed for Big Data, it’s possible to carry out an installation with storage possibilities in very large volumes and with incomparable performance on several server racks. This type of installation isn’t described here for simplification because it concerns experts in urbanization of Big Data infrastructures.

NOTE Hurence propose support to install data historian on server clusters (multi-server installation) for large volumes. It will also be able to help you switch your mono server installation to a multi-server installation.

Pre-requisites for a single server installation

The minimum server configuration for a single server installation of data historian is:

- an OS CentOS ou Redhat 7.x
- 64 bit architecture
- 32 Gigabits of RAM
- 32 vcores of CPU
- 2 Tera octets of disk
- Java VM Version 8

The Data historian’s components

The Data historian is composed of:

- a time series storage back-end. In a default installation of the Data Historian, this back-end is SolR (Apache SolR).

NOTE Contact Hurence if you require an another back-end like Elasticsearch or Open Distro.

- an API to query the back-end in REST. This allows in particular to integrate the Data Historian with visualization tools like Grafana (ou Qlikview ou autre)
- a tool for visualizing temporal data - **grafana** and a set of plugins for visualize the data contained in the back-end.
- batch injection components: these are a set of **scripts Spark** llowing massive data injections or simple imports of Excel or CSV files..
- Message bus - **Apache Kafka** - to receive real-time time series. We will see later that lots of tools can populate this message bus, including gateways from MQTT.
- a real-time analysis component - **LogIsland** . This component analyzes the time series circulating on the bus with raising of alerts or others and storage of the data in the back-end.

We will progress in a simple way and initially abandon the real time part. Then it will be added. Then we will explain how to make injection chains that will correspond to the real cases of companies, especially factories.

In this section, we will install in order:

- Java in openJDK 8 version
- the SolR back end SolR
- the gateway REST
- grafana
- Spark and scripts needed for simple and massive injections

IMPORTANT	This installation requires that you have internet access. If this is not the case, your administrator must provide you with the necessary tar.gz.

The Real Time Advanced Processing section will result in the following deployments.

Compatibility matrix

The Data Historian compatibility matrix is as follows:

In doubt, we advise you to take the latest stable version of each technology on which the Data Historian has been tested.

Install the pre-requisites

The following commands must be executed to install the prerequisites (Java). All orders are made with the user **root** unless other instructions are given.

For reasons of good practice we update the system before installing.

```
yum -y update
```

And we install java - with its JDK because we will need it later. We choose OpenJDK but JavaJDK or AdoptOpenJDK could be used.

```
yum install java-1.8.0-openjdk
```

Check that you have the correct version of Java by default with the command:

```
java --version
```

We will now install wget which will allow us to download the packages necessary for our installations.

```
yum install wget
```

Install the SolR back end

We will download the latest version 8.4.x (replace the x with the latest version of SolR):

```
wget https://downloads.apache.org/lucene/solr/8.4.1/solr-8.4.1.tgz
```

Move the tgz into / opt and unzip it:

```
mv solr-8.4.1.tgz /opt

cd /opt

gunzip solr-8.4.1.tgz

tar xf solr-8.4.1.tar
```

The documentation on SolR explains the content of this distribution. For the version that concerns us it is at this URL http://lucene.apache.org/solr/guide/8_4/installing-solr.html#directory-layout

Start SolR

Starting SolR is very simple just do

```
bin/solr start -force
```

WARNING

Of course it is not always a good idea to launch a tool as root. In theory we recommend creating a user **solr**. To give him the rights to the directory /opt/solr-8.4.1 and then to make a `su solr` before launching the start command with his login. In all cases, securing a platform of this type requires expertise, in particular to use the rights given in a compagny ActiveDirectory or other LDAP directory. Hurence can assist you in securing these tools.

You can verify that SolR has started with the command

```
bin/solr status
```

If you can open a browser on this server - replace localhost by the name of your server - on the port 8983 (if the port isn't blocked), the SolR management interface appears.

http://localhost:8983/solr/

Install the REST gateway of data Historian

Download the data historian tar.gz on the Hurence downloads site (make sure you are still in the **opt** directory):

```
wget https://www.hurence.com/Downloads/datahistorian/datahistorian-1.0.1.tar.gz
```

Unzip the data historian archive:

```
tar xvfz datahistorian-1.0.1
```

Launch the REST gateway:

```
bin/gateway start
```

Check the status of the gateway:

```
bin/gateway status
```

Inject a sample CSV file:

```
bin/import-csv samples/data-sax.csv
```

Check that the data has been imported with solR by going to the interface http://localhost:8983/solr/ and doing the following research:

Do this research, you see that the historian is able to give you research the somewhat identical time series in sequences of values. These are called semantic annotations. The values have been transformed into letters and the value sequences form "searchable words". The section on SAX encoding explains in detail what can be done with semantic encoding and its principle.

We are now going to install Grafana to visualize the data, put annotations and visualize alerts when we create them in the following sections.

Install Grafana

First you have to disable Linux security temporarily and then permanently by editing a file whose directives will be applied if the machine is rebooted in the future.

Disable linux security

```
setenforce 0
```

Edit the file /etc/sysconfig/selinux with the editor of your choice (vim or vi) and put SELINUX=disabled (instead of SELINUX=enabled)

Create a grafana repo by creating a file /etc/yum.repos.d/grafana.repo with vi or vim

Put in the following content then save the file:

```
[grafana]
name=grafana
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

Start the installation command:

```
yum install grafana
```

Install useful additional fonts:

```
yum install fontconfig yum install freetype* yum install urw-fonts
```

Start Grafana:

```
systemctl start grafana-server
```

Check its status which must be active:

```
systemctl start grafana-server
```

EPossibly have Grafana launch at boot with the command:

```
systemctl enable grafana-server.service
```

Grafana is now ready on your server on the port 3000 (remplace localhost by the name of your server): http://localhost:3000/