UNIVERSITÀ DI PISA

Artificial Intelligence and Data Engineering / Computer Engineering

Cloud Computing

Time series system based on Gnocchi

Project Documentation

Leo Maltese
Giulio Federico
Clarissa Polidori
Leonardo Poggiani

Academic Year: 2020/2021

Table of Contents

1 Introduction

We want to create a system that allows us to store time-series data in Gnocchi and then view them through Grafana.

The problem that **Gnocchi** solves is the storage and indexing of time series data and resources at a large scale. This is useful in modern cloud platforms which are not only huge but also are dynamic and potentially multi-tenant. Gnocchi takes all of that into account. Gnocchi has been designed to handle large amounts of aggregates being stored while being performant, scalable and fault-tolerant. While doing this, the goal was to be sure to not build any hard dependency on any complex storage system.

Gnocchi takes a unique approach to time series storage: rather than storing raw data points, it aggregates them before storing them. This built-in feature is different from most other time series databases, which usually support this mechanism as an option and compute aggregation (average, minimum, etc.) at query time.

Because Gnocchi computes all the aggregations at ingestion, getting the data back is extremely fast, as it just needs to read back the pre-computed results.

Grafana allows us to query, visualize, alert on and understand data also stored in Gnocchi.

The code of the various components of the service can be found on `https://github.com/CloudComputing-PosamanGroup/VallatiProject`

2 Service architecture

First of all it was necessary to deploy Gnocchi on our architecture manager. We then configured the controller to be able to connect it with Gnocchi through the REST interface that the latter exposes. So we were able to create an *archive-policy* in Gnocchi, which contained the metrics we wanted to insert. We later also created the *metrics* that could contain the necessary data.

2.1 Dataset

We used data that simulated the periodic insertion of new measurements inside Gnocchi, as if sensors were detecting them. For this purpose we used a dataset recovered on Kaggle (`https://www.kaggle.com/se18m502/bee-hive-metrics?select=weight_2017.csv`) which represents data collected on hives.

We used 4 different metrics:

- **flow**: For a date it contains the number of departures and arrivals from/to the beehive. A positive number indicates the number of arrivals and a negative number of departures. Note that this 2 values are in the data set with the same timestamp.
- **humidity**: Level of humidity through time of the beehive expressed in %
- **temperature**: Temperature of the beehive through time of the beehive in Cº obtained from 13 sensors
- **weight**: Weight of the beehive through time in Kg.

2.2 Architecture

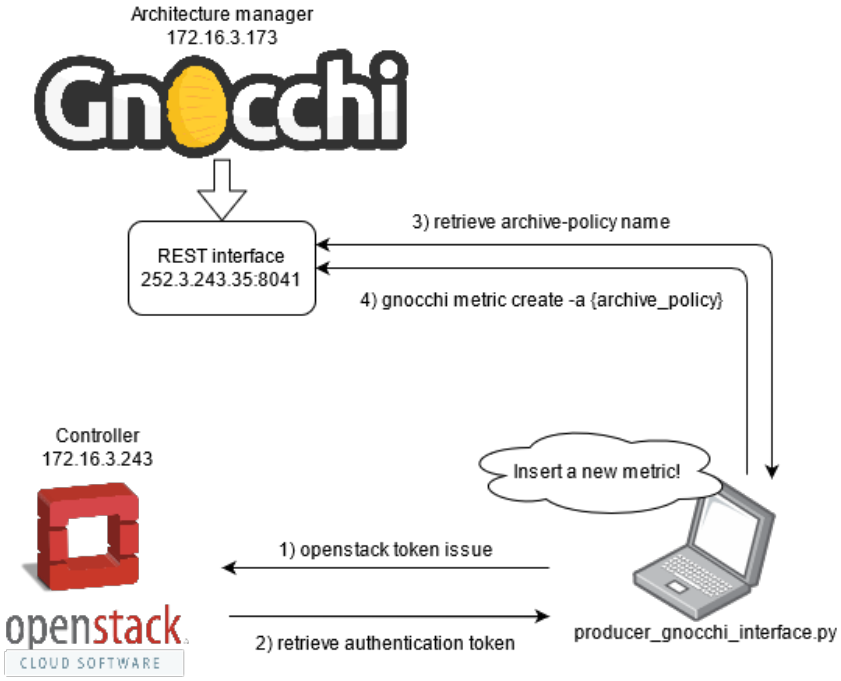

Figure 2.1: Service architecture

To access Gnocchi it is necessary to obtain the authentication token from Openstack and include it in subsequent requests to the REST interface that Gnocchi displays.

Gnocchi exposes a REST interface to create metrics, push the data and retrieve the data. The token can be retrieved through the following command:

```
# source adminopenrc.sh
# openstack token issue > output_token.txt
```

Requests to Gnocchi can be made via *curl* commands or via *gnocchi-client* or even through the python program *producer_gnocchi.py*.

The REST API usage can be retrived at the following link:
`https://gnocchi.xyz/rest.html`

The gnocchi-client commands can be retrived at the following link:
`https://docs.openstack.org/newton/cli-reference/gnocchi.html`

3 Grafana dashboard

To show the data contained in Gnocchi we connected the latter to Grafana through the plugin available online. In this way it is possible to configure Gnocchi as Grafana's data source and view the metrics through the metricID.

Grafana User Interface can be reached from browser web at:

- `http://172.16.3.243:3000/`

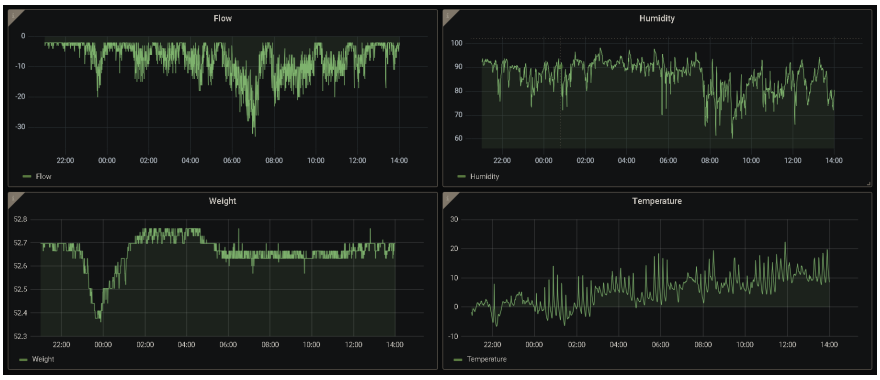

Figure 3.1: Static Grafana dashboard

Figure 3.2: Grafana dashboard real-time (1)

Figure 3.3: Grafana dashboard real-time (2)

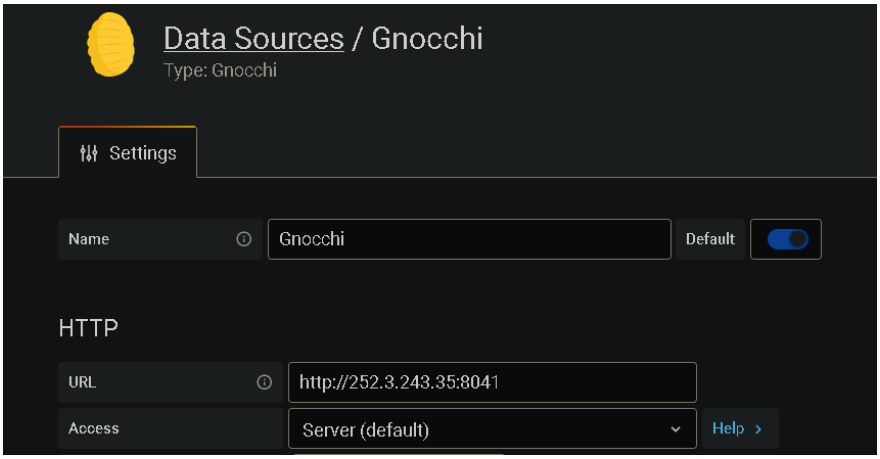

Figure 3.4: Gnocchi data source configuration (1)

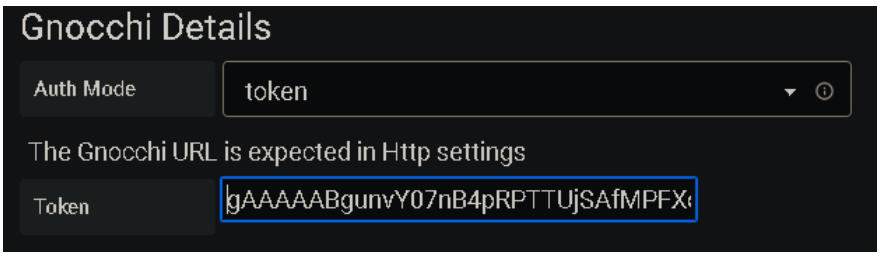

Figure 3.5: Gnocchi data source configuration (2)

4 | Python programs

We have written two simple interface programs in Python to perform operations on Gnocchi without going through the REST API. The first program, *producer_batch.py*, is responsible for authenticating Gnocchi by executing the commands that would normally be executed manually, taking all the data contained in the *data/* folder (they must be .csv files), creating a new metric for each of them and finally insert them as measures. Finally, the list of all metricIDs and also the authentication token is printed on the screen so that the user can enter these data directly on Grafana.

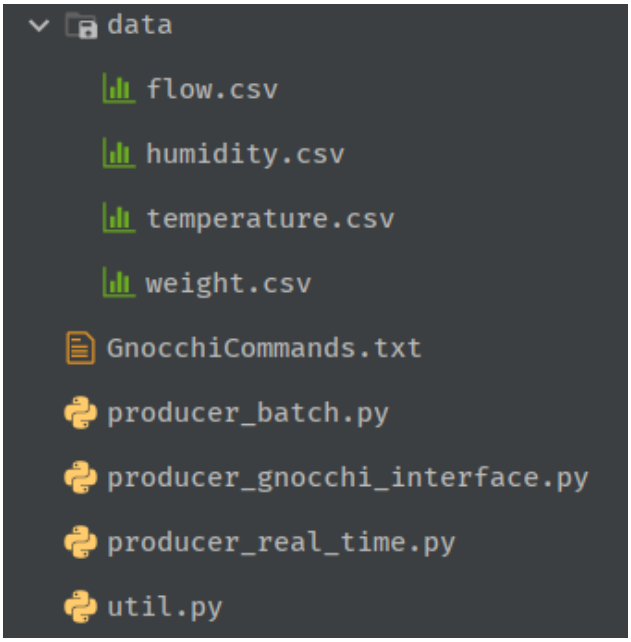

Figure 4.1: Python project structure

We have also created a simple command line interface that allows you to choose the operation to be performed on Gnocchi and to interface in a more user-friendly way (*producer_gnocchi_interface.py*). From the program it is possible to ask to see all the policies and metrics, create a new metric or destroy one, insert measures starting from a .csv file. Also in this case the authentication with Gnocchi is handled automatically.

Finally, it is also possible to simulate the behavior of a sensor with real-time data entry through the *producer_real_time.py* script. This program is used to insert data every T seconds in a cyclical way, for a maximum of points specified by the user.