University of Pisa

Laurea Magistrale (MSc) in Artificial Intelligence and Data Engineering

Project

Cloud Computing

Ceph-based File Manager

Academic year 2020-2021

Alessio Serra, Fabio Malloggi, Farzaneh Moghani, Marco Simoni, Valerio Giannini

Github: https://github.com/ValeGian/CC_Ceph_File_Manager

Project Specifications

Develop a **distributed file storage system** based on *Ceph*. The file storage must expose a *REST interface* through which external users can perform the following operations:

- **Retrieve** the list of files currently stored in the system
- **Delete** a file
- **Upload / Download** a file
- **Shows** current statistics on the status of the cluster

The application must be composed of two layers:

- **Frontend layer**, exposing a REST interface and receiving requests from clients
- **Backend layer**, exploiting the *librados python library* in order to interact with a *ceph-mon* module already deployed in a *juju container*

A different instance of the backend layer must be deployed on each of the three *ceph-mon* modules that are part of our installation.
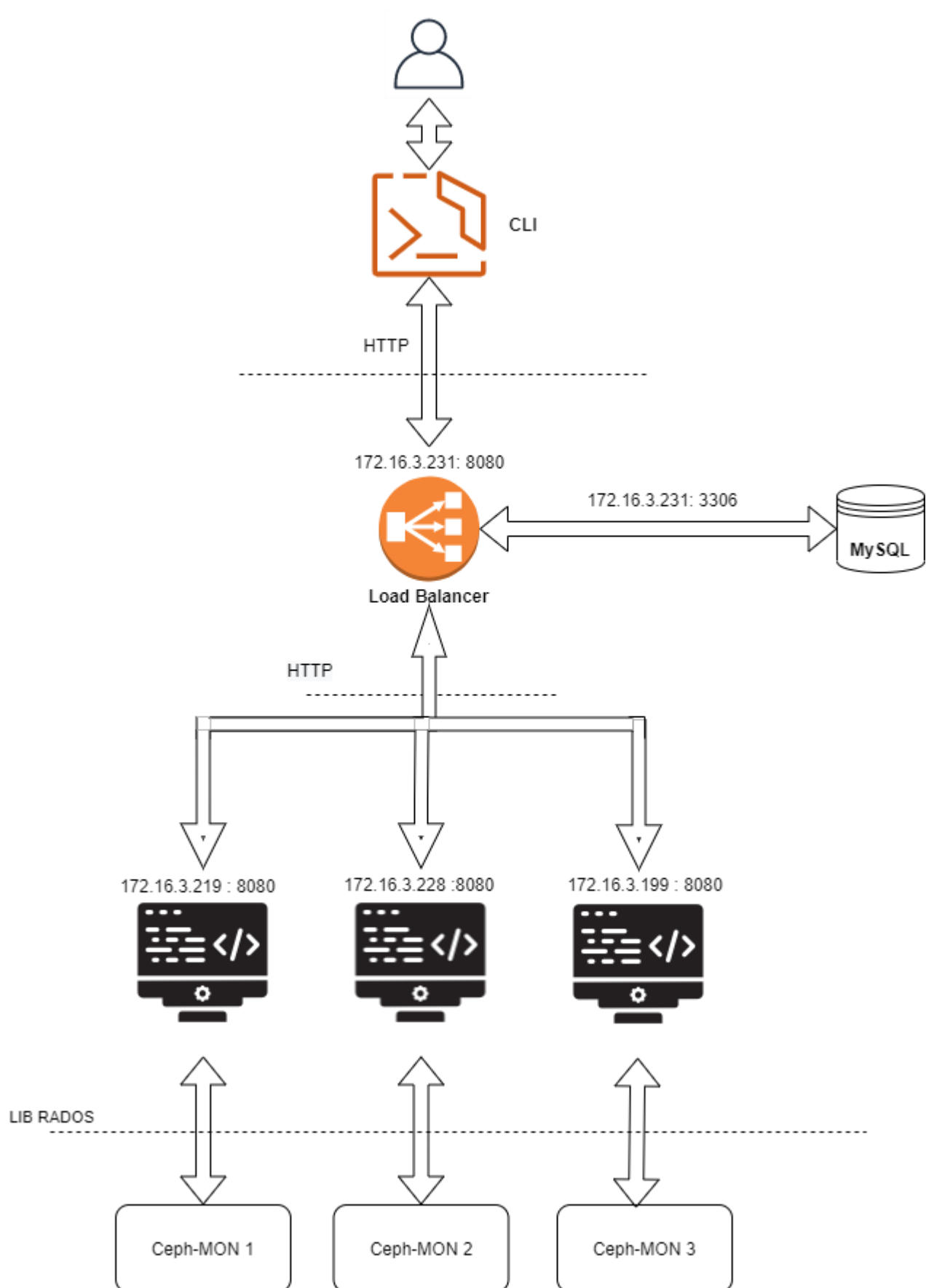
Architectural Design

The design of the application follows a **Service-Oriented Architecture** approach; each module exposes a *REST interface*. We exploit such approach to have a *synchronous communication* between **Client** and **Server**.

The **Frontend/Load Balancer tier** receives *HTTP* requests from clients and **forwards** them to the *REST API* exposed by the **Backend** module, which is then in charge of interacting with the *Ceph monitor* in order to provide an answer.

We deployed a **Load Balancer** module that acts as dispatcher of clients' requests. This module is in charge of balancing the requests between the various instances of **Backend** in order to increase resource utilization and ensure scalability.

Each time a request is received, the **Load Balancer** retrieves the list of **Backend** instances available at that moment from a **MySQL** database, which stores their IPs. This **MySQL** works as a **Shared Storage tier** and thanks to it we could easily update the state of the cluster, adding or removing **Backend** modules, to handle run time changes in the deployment' state.

We could easily deploy more **Load Balancer** instances that, by interacting with the **MySQL** database, would always share the same view of the **Backend** cluster.

CLI

HTTP
- -

172.16.3.231: 8080

172.16.3.231: 3306

MySQL

Load Balancer

HTTP
- - - - - - - - - - - - - - - -

172.16.3.219 : 8080   172.16.3.228 :8080   172.16.3.199 : 8080

LIB RADOS
- -

Ceph-MON 1   Ceph-MON 2   Ceph-MON 3

Frontend Design

We provide the following methods:

- **GET /objects** - to retrieve the list of the files
- **DELETE /objects/filename** - to remove a specific file
- **POST /objects** - to create a new file (the data of the file is included in the payload of the request)
- **GET /objects/filename** - to retrieve a specific file
- **GET /status** - to retrieve the current statistics of the cluster