A stateless + stateful web application

Group 01

Bellumat, Alberto <alberto.bellumat@studenti.unitn.it >
Gnutti, Alessandro <alessandro.gnutti@studenti.unitn.it >

May 2021

1 Introduction

Project Report made by Group 01 for the course project of **Fog and Cloud Computing 2021**.

2 The Architecture

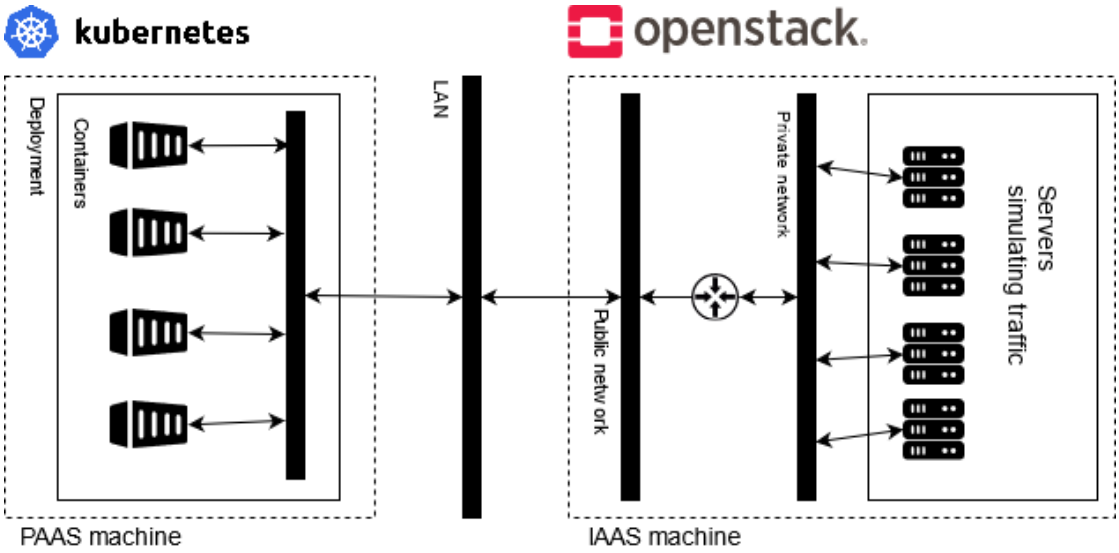
In the beginning, we thought of implementing an IRC architecture, where on the PaaS machine, we would have the IRC clients, and on the IaaS machine, we would have the IRC servers. This idea was conceived as some of us used to use IRC and as we found interesting articles about the IRC protocol. However, as we worked on building the IRC architecture we encountered more and more problems, in particular from the Open Stack and IRC servers side and, as time was getting shorter, it was decided to fall back to a more straightforward web architecture, taking also into consideration our experience in developing web architectures. In the web architecture, we wanted to host a website, make use of a MongoDB database and show how simulated traffic can affect the architecture.

On the PaaS machine, running the Kubernetes environment, we deployed a set of NodeJS containers that, organized in a deployment, deploy a simple website and a set of MongoDB containers, which will store the data sent through the website. The website has a simple form with two fields where the user can input data; when the data is entered, the list of data entered until that point in time is returned.

On the IaaS machine, we deployed a set of Ubuntu minimal machines, we call them "web_reqs servers", to simulate the traffic coming to our deployment from end users. The requests are made with a shell script that, using `curl`, automatically feed data to the form; the data inserted is the machine's name and a counter counting the number of the request. Via the `sleep` command we can make the script go faster or slower to simulate a lower or higher load. Each machine has its own volume for persistent storage and it is connected to a private networks which had its own router to connect to the public network.

Some of the code we employed to create our architecture can be seen at the following GitHub repository `https://github.com/dipperalbel/nodejs-cloud`.

The general architecture can be seen schematically in the diagram below:



The detailed structure of the Kubernetes cluster can be seen in the diagram below: