Final Project (A+B)

Goals

- Apply the knowledge obtained during the Cloud Computing and Virtualization MSc classes regarding, among which:
    - Provision Tools
    - Load balancing
    - High Availability
    - Service discovery
    - Scalability and Elasticity

Description

As approached extensively over this semester, to scale and take full advantage of virtualization and cloud computation, infrastructures (and apps) should be well planned and designed, able to scale horizontally when possible.

In this project you will need to propose an architecture able to scale an example website. The website to use is a simple PHP app provided by the professor (default installation). The project can be divided can be divided in two tasks or versions (A + B):

A) Use Vagrant and VMs to deploy and scale the application in a more robust and able to scale manner.
B) TBD – Use containers on premises or on google cloud to deploy the same app.

Deliverables

- Document describing the project, it should contain at least:
    - What was given to you? (O)
        - Including the initial architecture diagram of the application and infrastructure
        - Some metrics (hey, vegeta?)
    - What did you propose to improve the solution (separated A + B)
        - Proposed architecture (or ideal)
        - Description of the main components used, why you selected these, their pros and cons.
        - Some images and metrics demonstrating the solutions (O vs A vs B)
    - Configuration/code you find relevant to replicate / run.
- Slides and demo to present your solution (20m max)

Objectives

You have several directions and levels of complexity to aim for. While some might be harder to implement, you are advised to include them in your plan and indicate how it could be achieved and why you did not pursue the path.

Some possible ideas:

- Create a three layers cluster (load balancer, web, database)
- Serve the website correctly (partially or fully)
- Able to add extra nodes to the cluster manually (up + load balancer config)
- Detect new nodes and add them automatically to the load balancing layer
- Auto-scale web layer (e.g., between 1 and 3 depending on CPU load)
- High availability of the load balancer or other services
- Scale the DB layer (reads)
- Other (e.g., redundancy, deploy the solution using ansible, …)

Evaluation

- Report (20%)
- Quality of the solution / code (60%)
- Presentation (20%)

Note: All students must understand the solution they implemented.

Deadline

The project (relevant files + report) should be submitted around the exam date (TBD) (@ Moodle). Students are expected to present their solution in the following days (TBD).