

Final Project (A+B)

Project Goal

- By successfully concluding this project, students should have acquired knowledge on the various Cloud Computing and Virtualization topics, among which:
 - Provision Tools
 - Load Balancing
 - High Availability
 - Service Discovery
 - Containers and Cloud Computing
 - Scalability and Elasticity

Description

As discussed this semester, infrastructures (and applications) must be carefully planned and designed to be resilient and capable of horizontal scaling when possible to fully leverage virtualization and cloud computing technologies.

In this project, students must plan and implement two architectures that can scale and increase the availability of a given web app. The provided web app is a simple PHP project, currently running on a single VM (default installation under Vagrant - [repo](#)¹). It features typical web app functionalities such as static content, data loading from external APIs (cat pictures), JS libraries, sessions, file uploads, relational databases, and WebSocket communication.

It is recommended that students work in groups of **two**, but single or three-person groups are acceptable with justification. The professor **must** be informed of each group's constitution before the end of the contact period.

Challenge

The project is divided into two tasks or versions of the web app (A + B). Both have the same fundamental goal: start from the provided web app, analyze its architecture, plan/propose a newer architecture that uses cloud/distributed features, and implement it, justifying options and eventual limitations. Each version features a different approach and technologies:

- A) Use Vagrant and VMs (or similar, simulating on-premises IaaS) to deploy the application in a more robust, distributed, and able-to-scale manner.
- B) Use Docker containers (or similar, cloud-native solutions) on-premises or Google Cloud/Azure to deploy the same app. Use cloud services as desired for each part you find relevant.

¹ https://github.com/renatopanda/ipt_cloud_course_project
Cloud Computation and Virtualization

By accomplishing this and defending it successfully, students demonstrate a robust understanding of key concepts (high-level) and the ability to apply them independently of the solution and technologies. For example, understand the theory behind load balancing and use it through methods like a virtual machine and Nginx on-premises, an orchestration tool like Docker with Swarm/Kubernetes, or a cloud-native solution.

Deliverables

Groups must upload a ZIP file containing all the artifacts that you find relevant to explain and demonstrate your work. Among which:

- Document describing the project. It should contain at least:
 - What was given as base (O – Original app)
 - Including an analysis of the application and infrastructure's system architecture (with a diagram), technologies used, and identified issues.
 - Eventually, some performance metrics (e.g., using hey or Vegeta to see how many r/s it can handle)
 - The proposed solution (distinct sections for A and B)
 - Proposed architecture (the planned / ideal, and if very different, the implemented one) and what issues it addresses.
 - Description of the main components used in the implementation, why and how it was achieved.
 - Eventually, some metrics.
 - Brief discussion and conclusions
 - What are the key takeaways/lessons learned?
 - Some brief comparison between the three solutions (O vs A vs B)
- Configuration/code, instructions needed to replicate/run.
- Slides and demo to present your solution (15m max – can be delivered later)

Suggestions and Tips

Groups may have various goals and levels of complexity to strive for. While some features might be more technically challenging to implement, the students are advised to include and discuss such concepts in their report, even if they are not implemented in the final solution. In such cases, the report should briefly describe how those features could have been achieved and why the group decided not to pursue that path. In summary, groups are evaluated based on their proposed solution and critical planning/thinking, not only by what they demonstrate as working.

Below are some potential features and functionalities that the proposed solutions may include:

- A three-layer setup (load balancer, web, database)
- Being able to serve the website correctly (partially or fully)
- Ability to manually add extra instances to the cluster (or semi-automatically).
- Detect new instances and add them automatically to the load balancing layer.
- Auto-scale the web layer (e.g., between 1 and 3, depending on a schedule or CPU load)
- High availability of the load balancer or other services
- Scale the DB layer (reads)
- Storage solution to address the horizontal scaling and file uploads.

- Other (e.g., redundancy, monitoring, deploying the solution using Ansible, ...)

Evaluation

All students are required to attend the defense and have a clear understanding of the implemented solution.

Deadline

TO BE DEFINED DURING THE NEXT CLASS.