Cloud Computing Applications and Services

Project

2024/2025

General information

- Each group is composed of **5 elements**. Registration is done via the *Blackboard platform*.
- The project has **3 distinct assessment moments**. Two intermediate *checkpoints* with automatic assessment and a final delivery with in-person assessment. These are described in the **Assessment Section**.
- An FAQ with answers to questions and other important information for carrying out the work will be maintained and updated in the "Discussion" section of the ASCN-UM organization and will be accessible to all students.
- The repositories of each group will also be hosted in this organization.

Statement

The project aims to exercise the knowledge acquired in the course to automate the installation, configuration, monitoring, and evaluation of the **Moonshot** application (https://github.com/ASCN-UM/moonshot.git).

The Moonshot app is designed to manage the issuance, validation, and storage of Digital Green Certificates, which are used to record COVID-19 vaccinations, tests, and recovery status. The Moonshot system specifically serves as an application server for a mobile application, offering an application programming interface (API) for creating and managing certificates.

Tasks

Task 1 – Automatic Installation and Configuration of the Application

Objective: The first task is to use the **Ansible** tool to automate the installation and configuration of the Moonshot application on the **Google Kubernetes Engine (GKE)** of Google Cloud**.**

Requirements (*i.e.*, in addition to the use of Ansible and the GKE service)**:**

- The different components of the Moonshot application should run, when possible, in different *pods*.
- The installation and configuration of the application, as well as any service from the Google Cloud platform, must be carried out automatically in as few manual steps as possible.
- It must be possible to stop the application and run it again (*e.g.,* due to a scheduled maintenance task) without losing critical data (*e.g.,* user information).
- **It is mandatory** to respect the structure and functionalities of the provided Ansible *playbooks* (see the **Development Requirements** Section).

Result: Once installed and configured, the Moonshot application should be accessible from the outside (e.g., via a *browser* or *HTTP* requests*)*.

Task 2 – Application exploration and optimization

Objective: The second task aims to understand and optimize the application's performance, scalability, and resilience. The group is expected to answer the following questions:

1. Considering the base installation proposed by the group for Task 1:
    a. For a growing number of clients, which application components could be a performance bottleneck?
    b. How does the application perform against different numbers of clients and workloads?
    c. Which application components could constitute a single point of failure?

2. Given the answers provided to the previous questions:
    a. What load distribution/replication optimizations can be applied to the base installation?
    b. What is the impact of the proposed optimizations on the application's performance and/or resilience?

Considerations:

- The previous questions must be discussed in the project's final report, using the knowledge acquired in the course.
- The group must apply load distribution/replication mechanisms that allow for optimizing the scalability and/or resilience of the base installation.
  Note: To reduce the complexity of this task, the group should only focus on a **single component** of the application (*e.g.,* application server, database), justifying the choice.
- It is suggested that the group explore the monitoring tool the Google Cloud platform provides and consider different monitoring metrics (*e.g.*, CPU, RAM, I/O).
- Experimental tests should be considered to evaluate different functionalities and/or components of the application, considering the typical interaction of users with it.

The optimization, evaluation, and monitoring solutions proposed in this task must be **included in the Ansible *playbooks*** so they can be executed and reproduced automatically.

Assessment

The practical work includes 3 evaluation phases, which are described below:

***Checkpoint* #1 (2 values):** This checkpoint aims to evaluate the correct specification and functioning of the *Docker image* for the **Moonshot** application layer. **Deadline:** 27/10/2024

***Checkpoint* #2 (5 values):** This checkpoint aims to validate that, after installed and configured in the Google Cloud platform, the **Moonshot** application works correctly and is accessible from the outside. **Deadline:** 01/12/2024

Considerations:

- Checkpoints will be evaluated automatically through the *group*'s **GitHub repository.**
- For these checkpoints, it is <u>not</u> necessary to submit a report.

Final Assessment (13 points): The final assessment will consider all components of the project that were not automatically assessed at *checkpoints*. **Deadline:** 03/01/2025

Considerations:

- The project is **completed with the delivery of a report via the** *Blackboard* platform.
- **All scripts and configuration files** used to carry out the work must be made available through the **group's GitHub repository**.
- It is **mandatory** that the report identifies the group number and all its elements on the cover.

- The deadline for **submitting the work is January 03, 2025**. After this date, reports or new *commits* in the GitHub repository **will not be considered.**
- The presentation and discussion of the work will take place in the week between January 6th and 11th, 2025.

Report Requirements

The report submitted must contain the following information. Please note that the order in which each of the points below is presented in the report is at the discretion of the group.

- A brief description of the architecture and main components of the **Moonshot application**.
- Identification of the tools and approach used for the automatic installation and configuration of the application.
- A critical discussion of the questions posed in Task 2.
- Chosen monitoring tools, metrics, and visualizations, justifying your choice.
- Evaluation tools and tests developed, justifying your choice.
- Presentation and analysis of the results of the experimental evaluation.
- Final reflection on the work, pointing out the main strengths and areas for improvement.

The report does not need to include provisioning scripts (these are available in the GitHub repository) and should focus on the main decisions taken to address the points specified in this assignment. Although there is no maximum page limit, a 10-page report should be sufficient.

Furthermore, **all members of the group** must be prepared to give a brief presentation of the report, which includes a 5-minute demonstration. This demonstration must be carried out using Google Cloud, and for the sake of time optimization, each group must ensure that the *Kubernetes cluster* GKE is created and usable before the presentation.

Development Requirements

All scripts and configuration files used to carry out the work must be made available through a **GitHub repository** in the **ASCN-UM organization** (https://github.com/ASCN-UM).

Access to the organization will be given based on the information collected through the form https://forms.gle/NGJAKYvV5sYQEUdq7. Problems with access to the organization must be reported to the teaching team.

An ***FAQ*** will be maintained and updated in the “Discussion” section of the ASCN-UM organization, accessible to all students. This FAQ will contain information about:

- Details about the assessment for each *checkpoint*.
- Structure of the base code that each group must use to carry out its evaluation.
- Access to the Google Cloud platform.

Finally, the teaching team will provide a set of Ansible *playbooks* as a base that groups should use. They will be made available later and notified via the *Blackboard platform*.