1 Recipients

This assignment is intended for all students enrolled in the course Computação Distribuída e em Nuvem (CDN) who have opted for continuous assessment, which will take place during the academic semester. This choice becomes effective as soon as the student joins one of the working groups that submitted a solution proposal during this assessment period (via Moodle).

Students should form groups of up to three members in order to best divide the tasks defined in this assignment. The assignment represents 100% of the final CDN grade and requires a minimum grade of 9.5. Members of the same group may receive different grades.

The dates and procedures for each submission will be clearly indicated in Moodle.

2 Objectives

This project will serve as an integrative element of the knowledge acquired in the CDN course.

Each group/team should work on the topic defined in this statement, specifying and representing a set of functionalities. Although the topic is not open-ended, students should work towards completing the topic described.

Students must design and implement an asynchronous, queue-driven cloud architecture using AWS managed services and a private-network compute tier. All infrastructure must be provisioned via Infrastructure as Code (IaC) like OpenTofu or any other Terraform-compatible IaC.

The final system must demonstrate decoupling, event-driven processing, VPC subnet isolation, and autoscaling.

NORTE 2030 Programa Regional do Norte · PORTUGAL 2030 · Cofinanciado pela União Europeia

3 The problem

You are tasked with developing a highly scalable and fault-tolerant ticketing platform that mimics the functionality of a Ticketmaster-like website. The system allows users to request and purchase tickets for an event in real-time. The platform must be capable of handling sudden spikes in traffic, ensuring a smooth and seamless user experience during peak demand periods. Although the system represents a simplified "ticket distribution" workflow, it does not require the implementation of complex business logic.

The frontend will be publicly accessible, either through CloudFront or a simple static website hosted on S3. It should show a dynamic value received from the backend, such as "n people in front of you" or "request status" to give users real-time feedback about their ticket purchase progress.

The backend will run on ECS Fargate, with the service containerized for easy management. This backend service should be placed within a private subnet and only exposed to the internet through a secure private integration. This could be set up, for example, route requests through API Gateway, followed by a VPC Link to reach the ECS service. Alternatively, students can use a different method of their choice, as long as it maintains security and proper separation between the public and private components.

To ensure the application is properly segmented and secure, students need to set up at least one VPC with both public and private subnets. The ECS tasks should only run in the private subnets to avoid direct exposure to the internet. Any public-facing services, such as the frontend or API Gateway, should be routed through AWS-managed endpoints like CloudFront, API Gateway, or possibly an ALB (Application Load Balancer) if your account permits.

NORTE 2030 Programa Regional do Norte · PORTUGAL 2030 · Cofinanciado pela União Europeia

When it comes to scalability, students will need to configure horizontal scaling for ECS. The minimum number of tasks should be set to 1, but there should be the potential to scale up to 2 or more tasks depending on load. Scaling should be triggered by metrics like CPU usage, memory consumption, or incoming request counts. There is the need to justify which metric was chosen for scaling based on your system's behaviour.

Finally, all infrastructure should be described using OpenTofu. Your code should be functional and capable of being executed in a clean AWS account without any manual setup. You will organize your Tofu code into separate modules or files for different components: the VPC, ECS configuration, frontend, and IAM roles. The state of the infrastructure can be local or remote, but the configuration must be consistent and reusable.

4 Deliverables

1. Architecture Diagram
2. IaC (OpenTofu) Codebase
   - Must deploy the entire system;
   - Must include variables and outputs;
   - Code must be runnable with minimal manual intervention.
3. Any Dockerfile that was used to generate container images
4. Short Report (max 3 pages)
   - Summary of chosen architecture;
   - Justification of design decisions;
   - Instructions on how to deploy.

NORTE 2030 Programa Regional do Norte PORTUGAL 2030 Cofinanciado pela União Europeia

5 Report Preparation

This task consists of writing a report describing all the work carried out, showing and commenting on the results obtained, and presenting the respective conclusions.

There is no restrictions regarding the structure of this document.

6 Demonstration and Discussion

During the demonstration process, each group may use their own computer or one of the computers in the laboratory of the room where the course unit is held.

Each member of the group may be asked to answer questions about the contents of the course unit and, specifically, about its implementation in practical work. In addition, changes or the implementation of new features may be requested in order to demonstrate knowledge.

Members of the same group may receive different marks depending on their performance in the presentation and discussion.

7 Assessment Criteria

Each student's mark will be calculated based on three main components:

1. Individual performance during discussion;
2. Quality of the project;
3. Quality of the report.

The student's performance and knowledge of the content covered in the course unit will be assessed. The component of the report will be assessed based on the clarity, objectivity, and detail of the report.

8 Peer Review

Each group should conduct a collective analysis of the contribution and

NORTE 2030 Programa Regional do Norte | PORTUGAL 2030 | Cofinanciado pela União Europeia

effort of the human resources involved (which should be attached to the document representing the proposed solution). This analysis should make it possible to identify the contributions made by each member of the group and whether they are above, average, or below the contributions made by the other members.

9 Code of Conduct

The authors of the work must declare that they have acted with integrity and have not resorted to plagiarism, falsification of results, or any other practice that violates the code of conduct of the Polytechnic Institute of Porto.

Similarly, information (attached) on the use of Artificial Intelligence tools during the preparation of the document (if this is the case) must be made explicit, with reference to the nature of their use and the areas of the document that were affected.

10 Other Informations

The detection of fraudulent work, in part or in full, renders the assessment of such work invalid. In this case, the assessment of all members of the working group will be cancelled.

11 Notes for Students

- Backend logic can be trivial (e.g., return a counter, json status, timestamp, etc.);
- Simulate load using any tool (Locust, ab, curl loop);
- Minimal container image is enough; avoid unnecessary complexity.
- Use the project for demonstrating cloud reasoning, not for building a production system.

NORTE 2030 Programa Regional do Norte PORTUGAL 2030 Cofinanciado pela União Europeia