



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Runtimes for Concurrency and Distribution

Project: Exploration of orchestration challenges in a
microservice-based application

Luca Marchiori

June 2024

1. Report corrections
2. Take-home message
3. Learning outcome
4. CFU evaluation
5. Feedback on the course
6. More

My response to the "B" comments



To keep the report within the 10 pages limit, I had to cut some sections relative to theoretical definitions.

- Microservices definitions
- REST architectural style
- What are Swagger and OpenAPI
- Why choosing GO as programming language
- Alternative solution to the "latest" tag issue
- The choice of implementing common and basic microservices

Slides are available in the presentation but I prefer to focus on more important topics.

The focus of the presentation is on:

- Why virtualization, abstraction and orchestration
- How abstraction works in containerization
- The notion of business domain separation
- The separation between connection state and application state
- The problems encountered with the Horizontal Pod Autoscaler

Missing bibliography



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Take-home message



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



- **System design:**
- **Technologies:** Docker, Kubernetes

6 CFU equals to 150 hours of work.

How I spent my time:

- Lessons: ~ 48 hours
- Study: ~ 50 hours
- Project: 10 weeks, $\sim 6/8$ hours per week (60 hours)

Good balance between lessons and project, consistent with the CFU assigned.

Good points

- Very interesting topics that connect basic C.S. concepts with more advanced and real-world applications.
- I enjoyed the interactions between the students and the professor, especially for flipped classroom lessons.
- It would have been interesting to have more discussion sessions.

Improvements

- I felt a little lost during the project development. It is hard to start from scratch without a clear idea of what are the expectations.

Thank you for your attention!

“Microservices are **independently** releasable services that are modeled around a **business domain**. A service encapsulates functionality and makes it accessible to other services via **networks**” ¹

“A microservice is an architectural pattern that arranges an application as a collection of **loosely coupled, fine-grained services**, communicating through lightweight protocols” ²

[Back to index](#)

¹Sam Newman, Building Microservices

²Martin Fowler, Microservices

REST (Representational State Transfer) is a web architecture style using standard HTTP methods (GET, POST, PUT, DELETE) to interact with resources via URLs.

- POST: /users (create a new user)
- GET: /users/id (get a user by id)
- GET: /users (get all users)
- PUT: /users/id (update a user by id)
- DELETE: /users/id (delete a user by id)

[Back to index](#)

Microservices are language-agnostic, so they can be written in any language as long as they support necessary communication protocols.

It is also possible to have different microservices written in different languages as long as they can communicate with each other.

GO was chosen for the project because it has native support for web server making development easier and faster. ³

[Back to index](#)

³Thanks to the concurrency features of the language, and the native support for web standards, both Docker and Kubernetes are written in GO.

Solution to the "latest" tag issue



The "latest" tag leads to:

- Unpredictability
- Inconsistent updates
- Difficult in debugging and replication

Instead, use tags consistent with a version control strategy to ensure stability and traceability.

This approach allows for precise control over which image version is deployed, facilitating easier rollbacks and consistent environments across development, testing, and production stages.

[Back to index](#)

Why implementing common and basic microservices such as users, auth, and notifications?

- The goal was to learn about orchestration, not to develop impressive features.
- I didn't want to lose time by thinking of exceptional use cases.
- Those are component that I could find in any future project.

[Back to index](#)

What are Swagger and OpenAPI



Swagger

Swagger is a set of open-source tools built around the OpenAPI Specification that can help in design, build, document and consume REST APIs.

OpenAPI

OpenAPI Specification is a specification for building APIs. It is a language-agnostic specification that describes the structure of REST APIs.

In the project i used OpenAPI to define the API of the application and Swagger to have a GUI for reading the API documentation and test the endpoints.

[Back to index](#)