UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

WeatherWise – by WeatherWise team

A simple, but effective microservice weather web app.

course

Cloud Computing

author

The WeatherWise team

academic year

2023/24

Introduction

WeatherWise is …

simplicity! **all in one app!** **by users, for users!** **your weather app!**

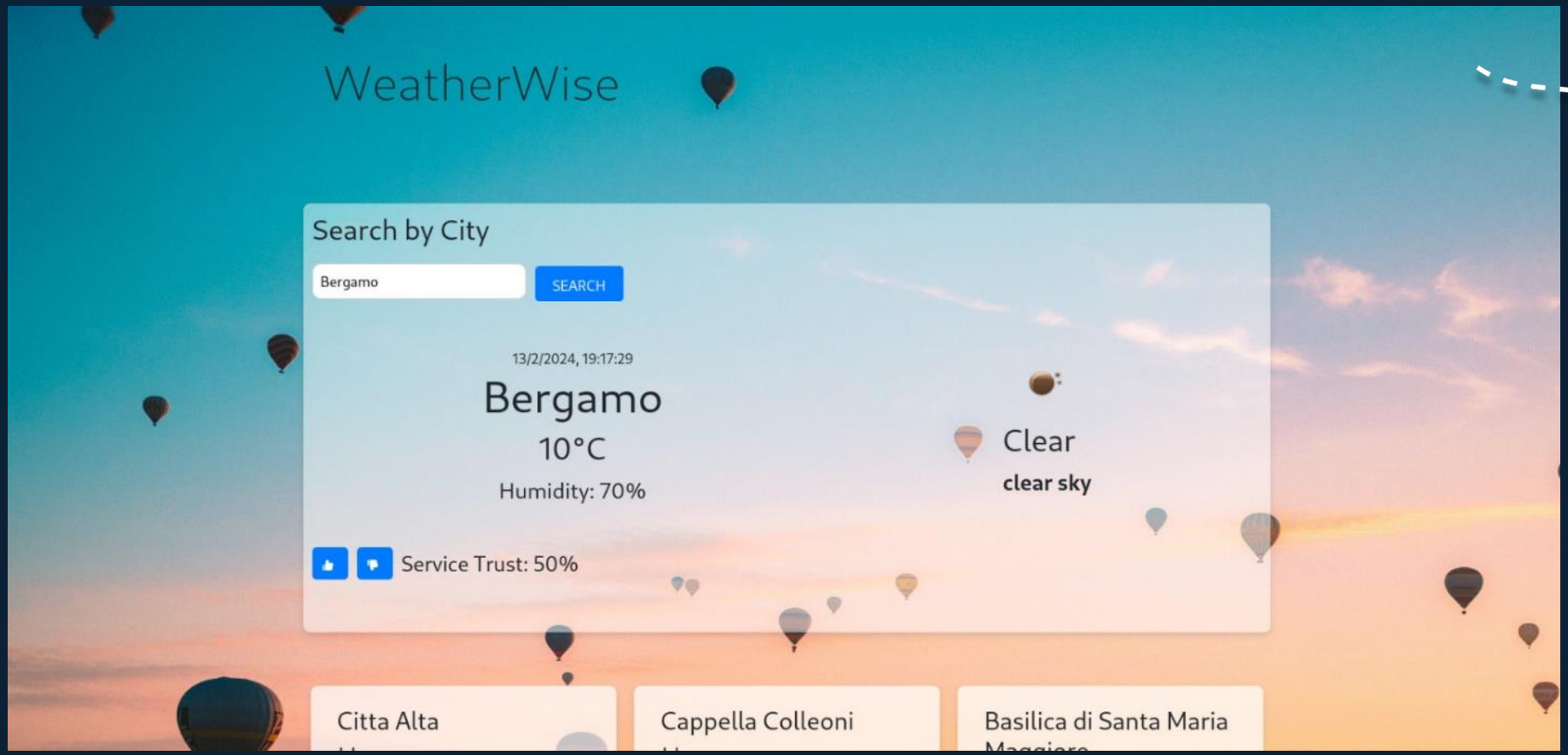
Functionality & implementation details

The first release of WeatherWise introduces some intuitive features for easy and user-friendly weather consultation for a given city, along with some interesting additional functionalities:

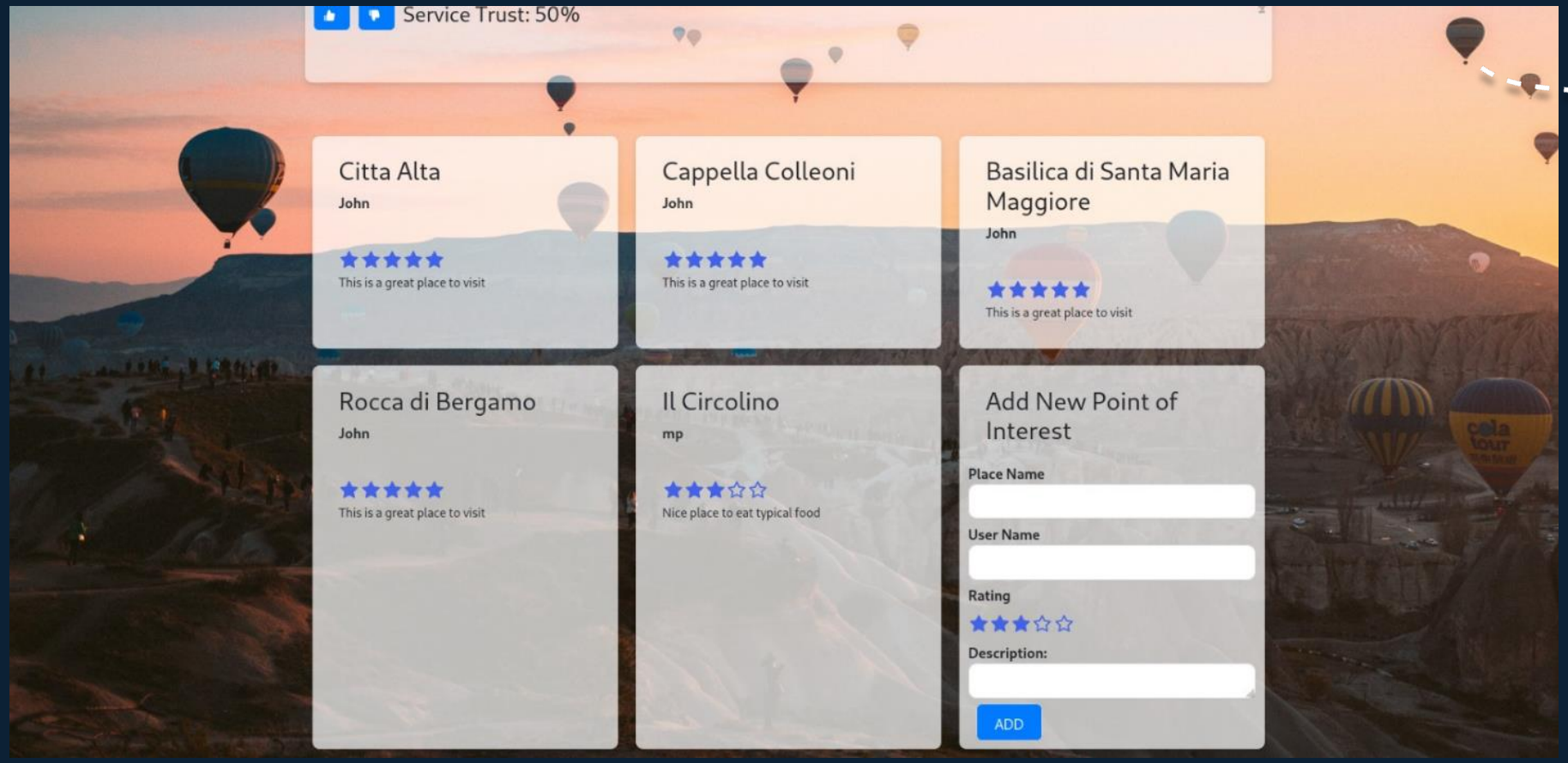
- Retrieval (GET) of real-time weather for the city, including temperature and humidity.
- Possibility to provide feedback (POST) for the current city weather forecast. The system allows associating a probability of accuracy for the forecast at that location (GET), based on feedback from the last 7 days.
- Possibility to provide advice (POST) on a "point of interest" to visit in the city, as well as getting an overview of all the places to visit in that location.
- The system allows controlled insertion of cities by providing an updated list of them (GET).

WeatherWise is...
more than a weather app!

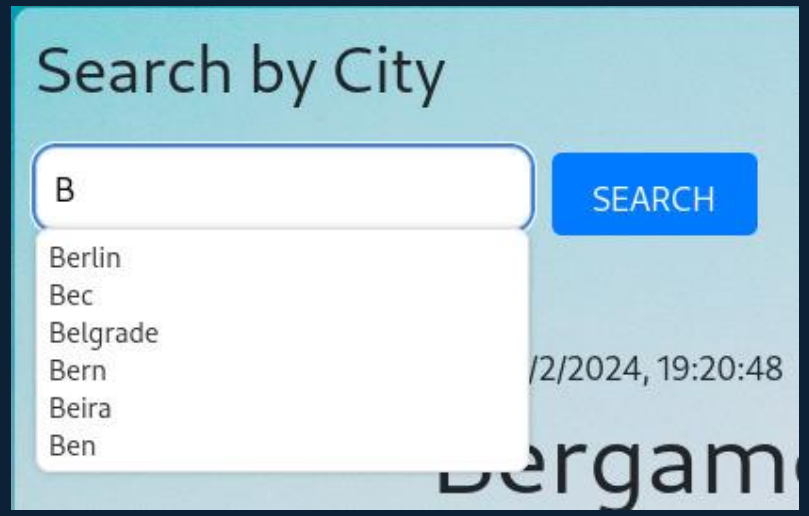
Functionality & implementation details



Functionality & implementation details



Functionality & implementation details



Microservices

The idea, compared to monolithic applications, is to divide the application into a series of smaller and interconnected services, autonomous and independently distributed software components (independent units of deploy).

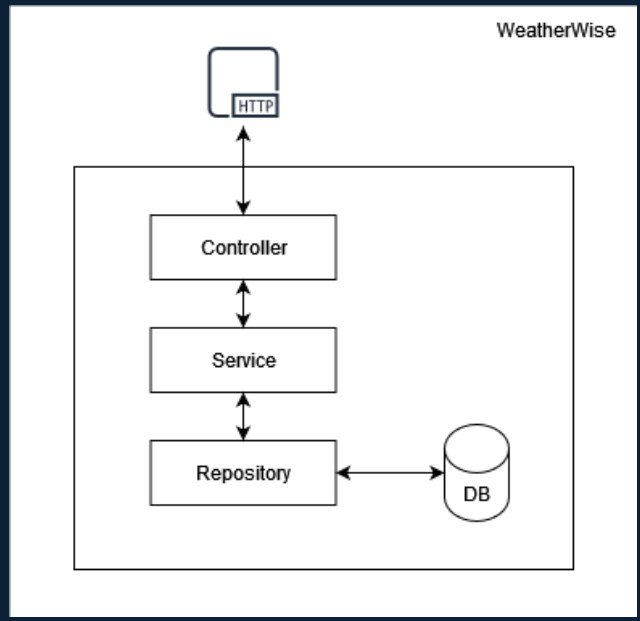
At runtime, each instance of a microservice is associated with a container or a cloud VM

Note.

- Microservices architecture addresses the problem of complexity;
- It allows each service to be developed independently by a team focused on it.

Some principles:

- Low coupling and high cohesion;
- Asynchronous is better than synchronous;
- Choreography is better than orchestration;
- Single Responsibility Principle.



Programming Languages, Frameworks and API

FRONT-END

- HTML;
- CSS;
- Javascript;
- Bootstrap;
- React;
- Axios.

BACK-END

- Spring;
- Spring-cloud;
- Java;
- JPA;
- Hibernate;
- Spring Data JPA;
- Javax Validation;
- Lombok.

DATABASES

- SQL;
- MySQL;
- PostgreSQL;
- FlyWay

WeatherWise has ...

more technologies **than cities in the database!**

API and API Rest interface

APIs (Application Programming Interface) are a set of definitions and protocols by which application software is built and integrated. A REST API, on the other hand, is a programming interface that uses HTTP to handle remote data.

In particular:

- REST stands for Representational State Transfer and it's an architectural style for distributed systems. It's, therefore, an abstraction, a design pattern, a way of discussing architecture without worrying about its implementation.
- HTTP, acronym for "Hypertext Transfer Protocol," is a communication protocol used for data transmission over the Internet. It's one of the fundamental protocols that enables clients (like web browsers) to request web resources from servers and receive them reliably.

API and API Rest interface

Microservices 1 – Weather Consulting

Base path: /meteo

Path	Method	Description
/{cityName}	GET	Return the weather forecast for the city, also providing information about humidity and temperature.

Microservices 2 – Places

Base path: /places

Path	Method	Description
/cities/{cityName}	**GET**	Return all the points of interest in a given city.
/	**POST**	Add a point of interest for a city based on the city's name, the name of the location, the user's name recommending it, a rating, and a brief description.

WeatherWise is...
RESTful!

API and API Rest interface

Microservices 3 – Feebacks

Base path: /feedbacks

Path	Method	Description
/percentage/{cityName}	**GET**	Return a probability regarding the accuracy of the weather based on the feedback provided in the last 7 days
/	**POST**	Add feedback on the effectiveness of the city's weather forecast.

Microservices 4 – City consultation

Base path: /cities

Path	Method	Description
/{prefixCityName}	**GET**	Return 5 cities that start with a given prefix.

WeatherWise is...
RESTful!

Microservices pattern

API-Gateway – A microservice Pattern

The API Gateway pattern is a common pattern in the realm of microservices, often used to manage client requests to a distributed architecture of microservices. In this pattern, an API Gateway serves as a unified entry point for all client requests. But what are the key points?

- Request aggregation;
- Request routing;
- Load balancing and failover.

Service Registry – A microservice Pattern

The "Service Registry Pattern" is an pattern widely used in the realm of microservices. This pattern is employed to enable distributed services to dynamically discover and communicate with each other in a scalable and highly distributed environment.

- A service registry is a central component that keeps track of available services in the network;
- Other services wishing to communicate with a registered service can query the Service Registry to obtain the contact information of the target service.

Other information

https://microservices.io/patterns/

Microservices patterns

Circuit breaker – A microservice Pattern

The "circuit breaker pattern" is a pattern that aims to improve the resilience and stability of the system. It is applied to manage calls between services, especially when there are external dependencies such as third-party APIs, databases, or other microservices. The pattern is implemented through three main states:

- Open;
- Closed;
- Half-open

Other information

https://microservices.io/patterns/

Config server – A microservice Pattern

The "config server pattern" is a pattern that allows for efficient and scalable management of configuration for distributed applications. In general, this pattern involves the use of a dedicated service called the "config server" that manages and provides configuration for different instances of microservices within the architecture.A service registry is a central component that keeps track of available services in the network;

- Other services wishing to communicate with a registered service can query the Service Registry to obtain the contact information of the target service.

External APIs

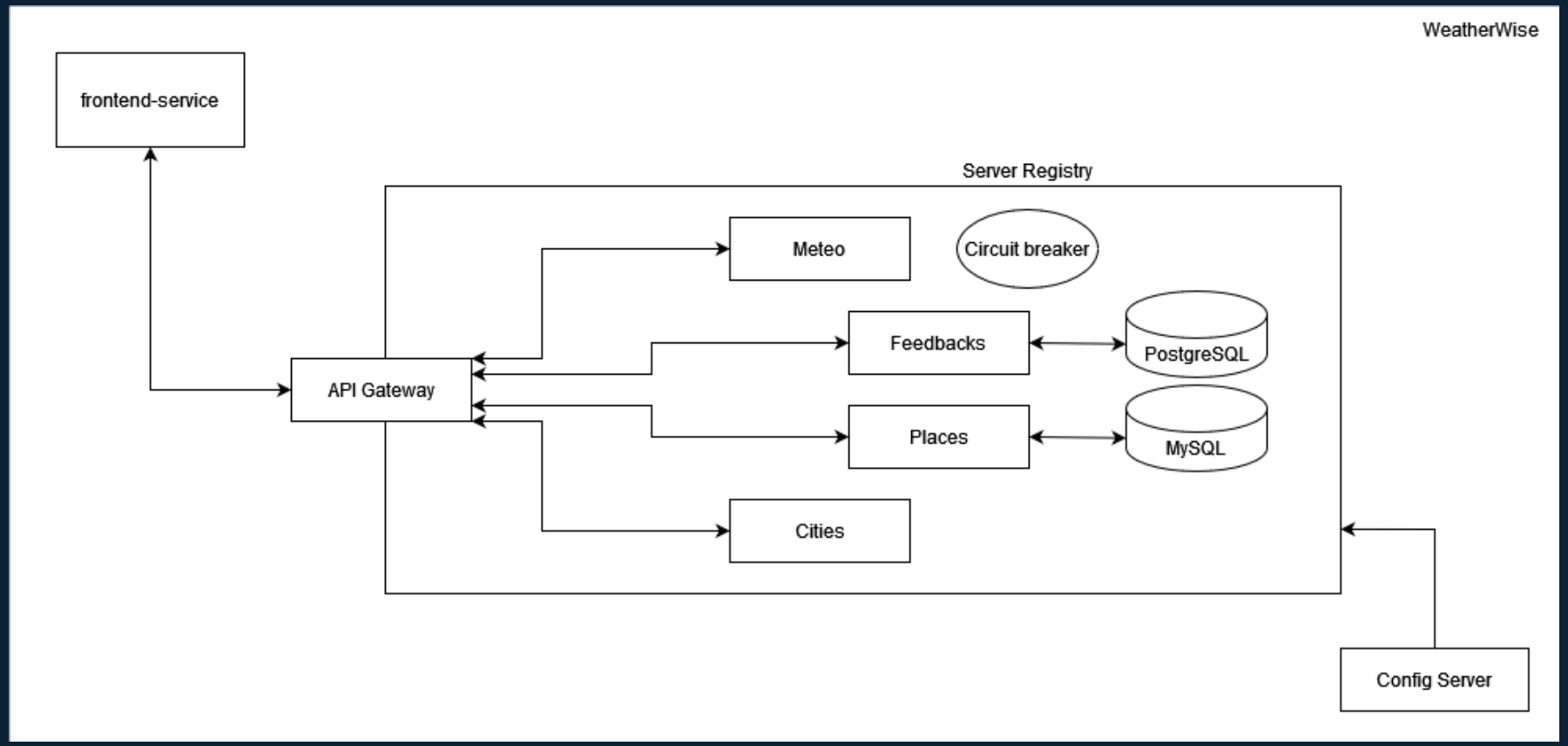
OpenWeather – some external APIs

OpenWeather is a weather forecasting service that provides real-time weather information and short- and long-term forecasts. It utilizes an extensive network of weather stations, satellites, and other sensors to gather meteorological data from around the world. These data are then processed using advanced meteorological models to generate accurate weather forecasts, including information such as temperature, humidity, wind speed, precipitation, and more. OpenWeather also offers APIs, used by us in WeatherWise, to allow developers to easily integrate weather forecasts into software projects.

Geonames – some external APIs

Geonames is a geographic database that provides information about geographical features of the world, such as city names, countries, rivers, mountains, lakes, and other geographic elements. Geonames also provides some APIs those allows developers to access the data and is used in WeatherWise for autocompleting the city name for which to make forecasts.

The architecture of WeatherWise



Containers

Containers provide an additional layer of abstraction and automation of operating system-level virtualization on Linux. The goal is to provide a consistent packaging and execution of software.

Containers and virtual machines have similar advantages in resource isolation and allocation, but they operate differently because containers virtualize the operating system rather than the hardware. Containers are more portable and efficient.

Within a container, all necessary executables, binary code, libraries, and configuration files should be present to run the application. However, multiple containers can run on the same machine, sharing the operating system kernel with other containers, each of them running as isolated processes in user space.

Benefits for WeatherWise

Each microservice is associated with a container in WeatherWise. This allows:

- Portability;
- Scalability;
- Resource Management;
- Agility in Development and Deployment;
- Security;
- Version Management.

A Cloud-Native Application

Kubernates

Kubernetes (K8s) is an open-source system, managed by the Cloud Native Computing Foundation, for automating the deployment, scaling, and management of containerized applications across multiple machines (called nodes).

Kubernetes makes global decisions about the cluster and detects and responds to cluster and application events. The control plane constantly monitors the cluster's state and reconciles differences between the current state and the desired state in response to events. K8s performs a number of tasks automatically, such as starting or restarting containers (or pods).

Benefits for WeatherWise

Why do we use Kubernates?

- Container Orchestration;
- Scalability;
- High Availability;
- Resource Efficiency;
- Extensibility.

WeatherWise has…
everything under control!

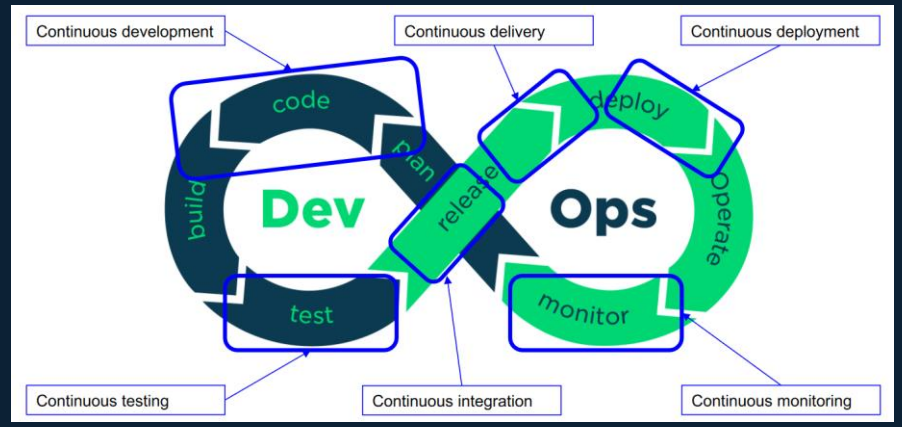
The DevOps Culture

The term "DevOps" doesn't have a truly concrete definition. It's a philosophy, a way of working, and means different things to different people. It's a software development method, a technology management approach that emphasizes communication, collaboration, integration, automation, and measurement of cooperation between software developers (dev) and operational staff (ops) in order to create and deliver software applications to their users.

Principles:

- Customer-centric action.
- End-to-end responsibility.
- Continuous improvement.
- Automate everything.
- Work as one team.
- Monitor and test everything.

DevOps has its lifecycle:



WeatherWise...

never stops!

Some tools

Minikube – your "mini-kubernates"

Minikube is an open-source platform that allows running a local Kubernetes cluster on a single machine. It is useful for developers and operators who want to test Kubernetes applications in an isolated and controlled development environment, without the need for a complete cloud infrastructure or a remote Kubernetes cluster. We used Minikube to run our application.

Docker HUB – a ship for everyone!

Docker Hub is a container hosting service that allows developers to distribute, manage, and collaborate on containerized applications. It provides both public and private repositories of container images that can be used as a foundation for creating and deploying containerized applications. It was used by the WeatherWise team to remotely store built images of services during the CI process.

GitHub action – a tool for automation

At WeatherWise, we have integrated the DevOps culture starting from creating a CI/CD pipeline to automate all processes. GitHub Actions is the service we used in WeatherWise provided by GitHub. It allowed us to automate workflows for our GitHub repositories, from building to packaging.

Other information

README, License, Authors e .gitignore

To conclude with the analysis present in this report, we would like to highlight the task of four files present in the project folder and which have, despite often being underestimated, great importance: README, License, Authors, and .gitignore.

- README is a file that contains information about the files contained in an archive or directory and is commonly included in software packages;
- License is a file that contains the full text of the license chosen for the project without any modifications;
- Authors is a file that identifies who worked on a particular project which critical for copyright management;
- .gitignore is a file in the git system that contains a set of items to be ignored by the version control system.

Released version

In software development, version corresponds to a certain state in the development of a software. Conventions for numbering a software version normally involve a triplet of numbers in the form X.Y.Z, where X, Y, and Z:

- X is the major version: which should increase only as a result of radical changes in the product, such as those that make it in some way incompatible with its earlier versions;
- Y is the minor version: which increases with the introduction of small features to complement existing ones, but maintaining substantial compatibility;
- Z is the patch version: which augments usually only by correcting errors with the same functionality.

We release the first version of WeatherWise (v1.0.0).

Other information

LICENSE

GPL-3.0 license

The GNU General Public License version 3.0 (GPL-3.0) is a software license developed by the Free Software Foundation (FSF). It is designed to ensure the freedom of users to run, modify and share software.

We summarize the key features of GPL licenses:

- Each program with its own source
- Each program must be accompanied by a copy of the licensed
- No constraints until you distribute
- If you make it publicly available, you also need the source
- As long as the source is there, it can also be sold
- No linking from closed applications

Benefits for WeatherWise

Why did we choose it?

- To preserve freedom. We want our software to remain open and free; GPL-3.0 is an excellent choice. It protects users' freedom and helps preserve the sharing and collaboration approach.
- Open development community. GPL-3.0 is often associated with open source development projects and communities that promote collaboration. By using this license, you can take advantage of the support and contributions of a large community of developers.

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

WeatherWise is...

HERE!

Thanks

The **WeatherWise** team

Mattia Piazzalunga, Nicolò Urbani, Matteo Severgnini e Davide Soldati

author

WeatherWise team

course

Cloud **Computing**

academic year

2023/24