Reactive Java on the cloud offers several benefits. Here are some interpretations by our team while working with Project Reactor as Java reactive framework. The application written with Project Reactor are packaged with docker and deployed on GKE.

Reactive programming allowed us to build more responsive, scalable, and resilient applications by handling asynchronous and event-driven scenarios efficiently.

Putting reactive on cloud provided following advantages:

1. Scalability: Project Reactor's non-blocking approach enables applications to handle a large number of concurrent requests, making it suitable for elastic cloud environments that require dynamic scaling. It simply means that based on the resources on hand, the application decides the concurrency and scaling. The concept of Schedulers.parallel & Schedulers.boundedElastic helps efficiently execute cpu intensive & short-lived blocking tasks, respectively.

2. Resilience: Reactor can handle failures gracefully by employing techniques like backpressure and circuit breaking, which are essential for maintaining system stability on the cloud. Constructs like onErrorContinue, onErrorResume and onErrorReturn help design the application resilience and handle failure scenarios persuasively.

3. Cost-effectiveness: Reactor applications can efficiently utilize cloud resources due to their responsiveness and scalability, potentially reducing infrastructure costs.

4. Microservices architecture: Reactive Java is well-suited for building microservices, which can be deployed independently in the cloud, allowing for easier maintenance, updates, and scaling.

5. Real-time data processing: Reactive Java can handle real-time data streams adequately, which is beneficial for various cloud-based applications, such as IoT, analytics, and monitoring. For real time data processing applications with high volume, Apache beam with Google Dataflow is being used. And for low volume we are using Project Reactor.

The reason of this decision and the comparison between the two needs a detailed explanation.

6. Enhanced user experience: Reactive Java's responsiveness and low latency can lead to improved user experiences, especially in cloud-based web applications.

Overall, using Reactive Java on the cloud can help you take full advantage of cloud computing capabilities and build robust, high-performance applications that can handle the demands of modern cloud-based environments.