Mass Market Billing Goes Cloud Native

Degree programme : MAS Information Technology

How can a file-based batch job application migrate to a cloud native state and benefit from its isolation, redundancy, resiliency and scalability? This essay investigates a container adoption for Swisscom's Output Management regarding the production of invoices and looks at the conditions under which a monolith legacy IT system can be transformed into a modern cloud native environment.

Olaf De Baets
olaf.debaets@gmail.com

Objectives

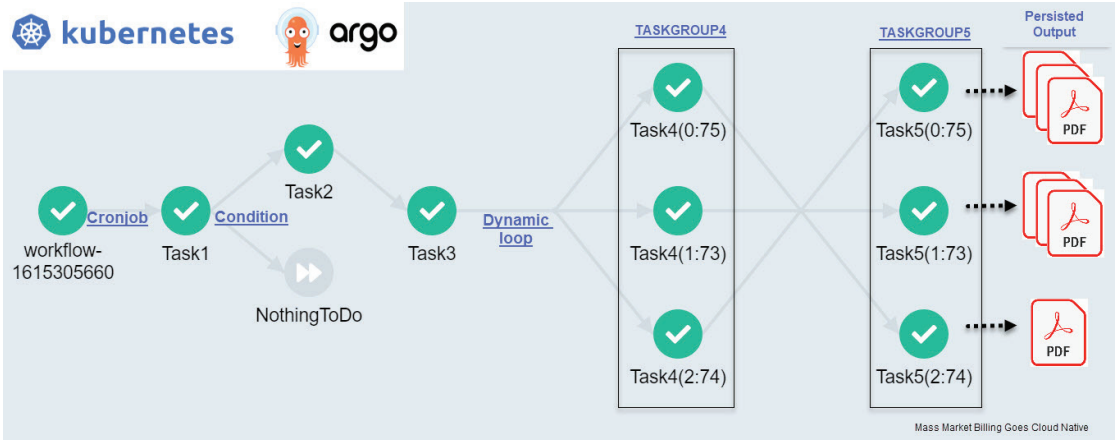
Cloud native computing permits access to quasi-limitless computational resources, but it needs to be demonstrated that it is also an effective technology in producing invoice documents, with its typical batch jobs of aggregating and formatting functions. With the help of containers and the Kubernetes orchestrator, an auto-scalable ecosystem for the billing application is made available as processes can run sequentially and in parallel. An automated workflow or an automated chain of decoupled processing steps needs also to be developed. This automation achieves the logical separation and the concealment of the Kubernetes abstraction from the application layer. These characteristics could generate in the future new business cases and opportunities.

Strategy

A comparison is made between different cloud native software products. Every pod that is scheduled and orchestrated constitutes an atomic and decoupled transaction and is part of a scalable and automated workflow.

Conclusion

The essence of this essay is that applications that are file-based and that do not have the necessary Webservices can be brought to a cloud native stage with the use of Kubernetes and Argo. Patterns such as „separation of concern" allow for decoupling and thus the possibility to introduce Kubernetes ReplicaSet and Deployment objects in the future. In the context of Swisscom's „Journey to the Cloud" even if it sometimes might take longer, it could be strategically important to benefit from the new capabilities. While doing this, we would typically assume a gradual approach in refactoring applications rather than a big-bang (redesign everything at once). Therefore, on this process of refactoring, part of the application would be brought to the new state with the new architecture where other parts of the application would remain as they were. Over time, we would migrate all parts of the application and end up with completely refactored application running as cloud native. The company would resemble the ship of Theseus in ancient Greek mythology. The ship was rebuilt little by little, replacing old planks by new ones, while it never stopped sailing its many seas. After several years, the entire ship appeared to be renovated. But was it still the same ship? To this deep question of identity, some say yes, others no. Swisscom could say both.



A workflow of pods orchestrated by Kubernetes and Argo