

PhD position

Modeling and Studying Self-Stabilization within Kubernetes

Hélène Coullon & Jacques Noyé
IMT Atlantique, LS2N, Inria, Nantes, France

Abdelhadi Chari & Pierre Crégut
Orange Labs, Lannion, France

Starting date Sept./Oct. 2023

Keywords: Kubernetes, Custom Resources, Operators, Deployment, Self-stabilization.

1 Context

This PhD position is funded in the context of the joint research team between Inria and Orange Labs: *STACK*². The joint team aims to work on operating large scale geo-distributed infrastructures such as the ones encountered in Fog or the Edge computing. In this context a subpart of the team work on automatic/autonomic deployment and reconfiguration of massively geo-distributed ICT infrastructures, and in particular on the promising aspects in using container orchestrators such as Kubernetes.

In fact, Kubernetes is nowadays not only used to easily manage containers of end-users applications, but also to manage large ICT infrastructures. This is made possible because network, storage and computing resources are nowadays all service-oriented to enhance flexibility and optimization capabilities. This leads to the current trend to move towards a fusion of ICT infrastructures and service management, with Kubernetes (as other containers orchestrators) as a promising candidate in this direction.

In Kubernetes, applications are implemented as containers working together. The structure of an application is described by an extensible language of resources. A resource is typed with a *kind* and is defined by a JSON structure following a schema associated to the kind. The behaviour of a resource is defined by another Kubernetes application : a *controller*. A controller tries to reconcile the expected state (described declaratively in the *spec* field of the resource) with the current state of the system (other resources, state of components external to Kubernetes) and iteratively modifies those resources until they coincide with the expected specification. The *status* of the resource is updated to provide feedback to humans and other controllers. Other mechanisms exist in Kubernetes that can modify or prevent resource definitions (conversion or admission *webhooks*). All those mechanisms are implemented as Kubernetes applications. Finally, the language is extensible: new kinds with their associated schema and controllers can be defined.

The Kubernetes promise is that although each controller looks only at a specific kind of resources and reconcile each resource independently of the others, at some point the system will converge toward a stable state that fulfills the expectation of all resources. Unfortunately each reconciliation action may disturb the state of other resources either generating large global oscillations of the global state or generating a state that cannot evolve and that does not fulfill the expectations (*e.g.*, admission webhooks run in the wrong order may destroy resources without any kind of recovery path). Finally, even when global convergence is guaranteed, it may be obtained from cascading small local convergence loops and be prohibitive in the number of reconciliation steps.

2 Work

Various mechanisms exist to alleviate those problems: explicit dependencies or notions of synchronized waves in GitOps frameworks, using init containers for explicit waits. But they require a lot of expertise both on the applications deployed and on the Kubernetes model to be used efficiently.

In this PhD offer we want to tackle the above problems through the following tasks:

- Understanding in deep detail how to write custom resources and their associated controllers, how to compose them.
- The formal modeling of Kubernetes custom resources and controllers, as well as their interactions.
- Studying from this modeling the self-stabilization mechanisms in Kubernetes.
- Understanding how to offer some guarantees on self-stabilization when using a set of custom resources.

To address these objectives a large study of the literature will be required, including: dynamic reconfiguration languages and frameworks that adopt the opposite approach with explicit dependencies [4, 5, 3]; workflow-oriented solutions of the DevOps community (*e.g.*, argo CD¹); approaches adopting distributed algorithms such as Consensus [1]; self-stabilization algorithms [2]; formal verification on self-stabilization of distributed systems².

3 Expected skills

The following skills are expected from the successful candidate:

- A Master's degree in Computer Science (or in the last year of an engineering school with a computer science option).
- Ideally, some knowledge of scientific research methodologies.
- Knowledge and experience in DevOps approaches such as orchestration with Kubernetes.
- Knowledge and experience in distributed software systems, in particular microservices.
- A good level of programming in Python or Rust, for instance.
- A good level of English to contribute to writing and presenting research papers.
- An ability to collaborate and communicate.
- Curiosity and an appetite for learning.

4 Additional information

Advisors

- [Hélène Coullon](mailto:helene.coullon@imt-atlantique.fr), IMT Atlantique & Inria & LS2N, Nantes, France, helene.coullon@imt-atlantique.fr
- Jacques Noyé, IMT Atlantique & Inria & LS2N, Nantes, France, jacques.noye@imt-atlantique.fr
- Abdelhadi Chari, Orange Labs, Lannion, France, abdelhadi.chari@orange.com
- Pierre Crégut, Orange Labs, Lannion, France, pierre.cregut@orange.com

¹<https://argo-cd.readthedocs.io/en/stable/>

²<http://www-verimag.imag.fr/~altisen/PADEC/>

Duration 3 years

Salary €2051 gross per month (year 1 & 2) and €2158 gross per month (year 3)

Location 70% at IMT Atlantique, LS2N in Nantes, 30% at Orange Labs in Lannion

References

- [1] Abdelghani Alidra, Hugo Bruneliere, H el ene Coullon, Thomas Ledoux, Charles Prud'Homme, Jonathan Lejeune, Pierre Sens, Julien Sopena, and Jonathan Rivalan. SeMaFoR - Self-Management of Fog Resources with Collaborative Decentralized Controllers. In *SEAMS 2023 - IEEE/ACM 18th Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2023. doi:10.1109/SEAMS59076.2023.00014.
- [2] Karine Altisen, St ephane Devismes, Swan Dubois, and Franck Petit. *Introduction to Distributed Self-Stabilizing Algorithms*. 2019. doi:10.2200/S00908ED1V01Y201903DCT015.
- [3] Maverick Chardet, H el ene Coullon, and Simon Robillard. Toward Safe and Efficient Reconfiguration with Concerto. *Science of Computer Programming*, 2021. doi:10.1016/j.scico.2020.102582.
- [4] H el ene Coullon, Ludovic Henrio, Fr ed eric Loulergue, and Simon Robillard. Component-based distributed software reconfiguration: A verification-oriented survey. *ACM Comput. Surv.*, may 2023. doi:10.1145/3595376.
- [5] Simon Robillard and H el ene Coullon. SMT-Based Planning Synthesis for Distributed System Reconfigurations. In *FASE 2022 : 25th International Conference on Fundamental Approaches to Software Engineering*, 2022. doi:10.1007/978-3-030-99429-7_15.