LTU Project, Declarative Edge Applications

stefan@avassa.io, 0705233262

Introduction

Applications are being pushed to the edge for several reasons, such as latency, privacy, and autonomy. Examples of the edge include robots in factories, self-driving trucks, and Edge AI video analytics. The container standard and the availability of Linux flavors for various platforms have enabled the deployment of edge applications. Edge is different from the cloud in many ways: the cloud is few places, edge is thousands of places, the cloud has unlimited resources, the edge has limited resources. Efficient orchestration of edge applications requires a declarative way of defining the edge applications and their requirements.

Avassa.io provides a platform to deploy container applications at the edge. The platform installs a small agent on each host to manage the container lifecycle and the infrastructure on the edge site. A central component in the cloud provides application deployment and central monitoring. Avassa has also defined a domain-specific language in YAML to declaratively specify the edge container applications. See more on https://avassa.io/

The *overall goal* and benefit of the outlined project is to simplify the authoring of declarative edge application specifications.

Problem statement

There are several well-known languages to define cloud applications, for example Helm Charts or the simpler and more limited Docker Compose. The problem to study in this project proposal is:

- Mapping between Avassa Application specifications versus Helm Chart and Docker Compose
- How do we simplify the authoring of Avassa Application specifications?
- How to convert existing Helm Chart and Docker Compose into Avassa specifications

Toolbox

- The Avassa language exists as an OpenAPI spec: https://avassa-api.redoc.ly/
- An authoring environment is most likely based on using a language server backend: https://microsoft.github.io/language-server-protocol/
- Generative AI to generate skeletons based on requirements

Output and goals

The work should result in:

- Mapping definition between Avassa Application specifications and Helm Charts, Docker Compose. (There is not a one-to-one mapping, will be incomplete/heuristic)
- Authoring front-end for Avassa Application specifications, VS Code and Web client
- Import tool from Helm Charts, Docker that will generate a skeleton Avassa Application

The project

The primary contact persons will be
Stefan Vallin, stefan@avassa.io. Stefan is the product lead for Avassa and works from Skellefteå.
Demo videos of Avassa: https://avassa.io/product-demo-tour/