Bare Metal K8s-Lab for Teaching

Degree programme: BSc in Computer Science | Specialisation: Digital Business Systems
Thesis advisor: Prof. Rolf Jufer
Expert: Dr. Joachim Wolfgang Kaltz

Deploying containerized microservice applications on Kubernetes platforms is an increasingly important topic for every digital business and has as such also found its way into the BFH curriculum. The goal of our thesis is to provide an automated solution to the challenges of deploying and maintaining a compact on-premises Kubernetes provisioning-solution for professors and students to use as an infrastructure option.

Introduction

In recent semesters microservice architecture has increasingly been discussed in theory in several BFH-TI modules. For most of the practical exercises or show cases, Kubernetes is used as infrastructure, providing a scalable environment for the aforementioned microservice applications. As of now, to run Kubernetes as infrastructure, mostly manually installed and configured solutions have been used. A cost-effective and flexible solution to provide Kubernetes clusters, which subsequently can be used for educational purposes or as infrastructure for school projects spanning over one or more semesters, is therefore currently not available for the degree programme.

Goals

This thesis is intended to provide the basis for the mentioned purposes. Dedicated Kubernetes clusters get requested over an internal contact (usually an assistant of the computer science faculty with higher education in electronics or computer science) and deployed with little manual effort. The Kubernetes API of each provided cluster will then be reachable from the internal BFH network. Furthermore, a cluster provides means to create persistent storage and all web traffic to and from deployed applications is SSL encrypted.

Technologies

On the provided single physical server, OpenNebula in combination with the hypervisor KVM is used to manage and create the virtual machines needed to run Kubernetes clusters. For all deployment, installation, and maintenance tasks of the solution we use Ansible as automation software. For the core Kubernetes functionalities, the distribution K3s in combination with kube-vip and MetalLB have been evaluated since the current hardware limitations put lightweight distributions in favour.

Implementation

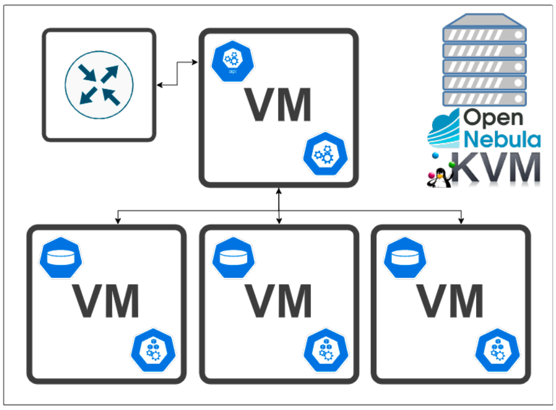
To be able to recreate or expand the entire solution, the automation has been split into four main Ansible playbooks: the automated installation of all software packages including the base configuration of OpenNebula with all templates, the configuration of a virtual router, the configuration of a NGINX reverse proxy to provide SSL encryption, and the provisioning of an entire Kubernetes cluster with all configurations. To provision a dedicated Kubernetes cluster, the person in charge will be able to execute one single Ansible playbook and handover the corresponding configuration file to the requestor.

Conclusion

With the provided requirements and resources, a stable and reliable Kubernetes provisioning solution has been built. Some aspects do not fulfil current best practices but for the intended use case all functionalities have been implemented. With the detailed documentation, the product can be further developed, and hardware resources expanded.

Luca Beyeler

Robin Füglister



Visualization of the bare metal server architecture