Lab Topology Builder K8s Operator

Implementation of a Kubernetes operator to manage the deployment of emulated network topologies

Graduate

Jan Untersander

Tsigereda Nebai Kidane

Initial Situation: Lab Topology Builder (LTB) is an application developed by the Institute for Network and Security (INS) and is used for research and teaching purposes. It can create networking labs, which are emulated network topologies with multiple interconnected nodes (servers, switches, routers, etc.). It is a key component of multiple courses at the OST and is used by students to practice and learn about cloud, networking and security concepts. Currently, the application is based on a React frontend and a Django Python backend in combination with KVM/Docker for the deployment of the labs. This solution has grown organically over the years, with multiple contributions from term projects and bachelor theses. Maintaining the application has become increasingly challenging, because of technical debt and missing documentation. Therefore, making it open source would require a substantial refactoring effort.

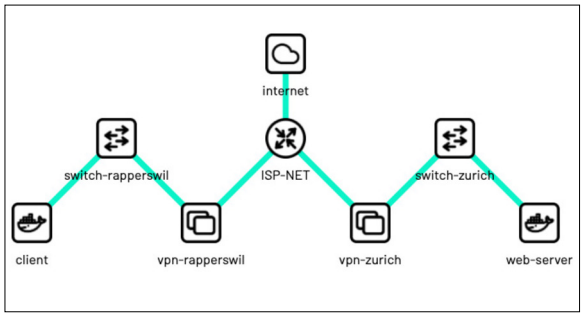
Objective: With the overall goal of open-sourcing the application, the objective of this project is to create an LTB-inspired Kubernetes operator, which can be used to deploy and manage the aforementioned labs on a Kubernetes cluster. A Kubernetes operator offers automation, dynamic resource allocation, and seamless integration with a wide range of tools. Thus, making it ideal for managing complex applications while enabling effortless scaling and orchestration. The operator should be able to manage labs consisting of pods with containers and KubeVirt virtual machines and should accept a lab definition in the form of a YAML file. The different lab nodes should be able to communicate with each other via a layer 3 connection. The operator should also provide a feature to query the status of a lab. Additionally, the lab nodes should support out-of-band management through various protocols. Optionally, access to a lab should be restricted to specific users using access control policies.

Result: The project resulted in a Kubernetes operator for the LTB application, which takes three different custom resources (CRs) as input:

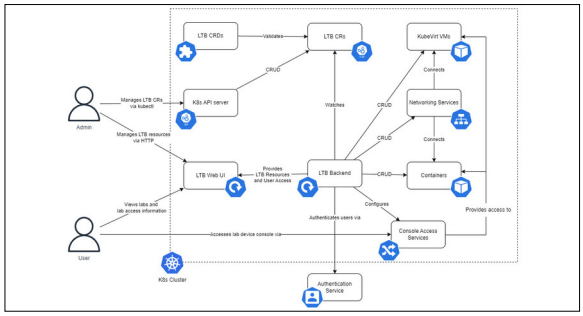
- Lab templates
- Lab instances
- Node types

The lab templates define the lab topology i.e. the connection between the nodes and the configuration of the nodes. The node types define the different types of nodes that can be used in a lab. Finally, the lab instances are the actual labs that are deployed. Lab instances reference a lab template, which in turn references node types.
There is support for layer 3 networking between the lab nodes limited to a single physical Kubernetes node. Access to the out-of-band management of the lab nodes is provided via a web-based terminal or a freely configurable port which will be forwarded to the lab node. Information regarding the lab's status and remote access details can be obtained via a command-line interface (CLI). Although access control has not been implemented, a groundwork has been laid by separating lab instances into their own namespaces. This allows for the use of Kubernetes' built-in role-based access control (RBAC) to restrict access to lab instances.
In summary, the LTB Kubernetes operator enables users to deploy and efficiently manage labs within a Kubernetes cluster.

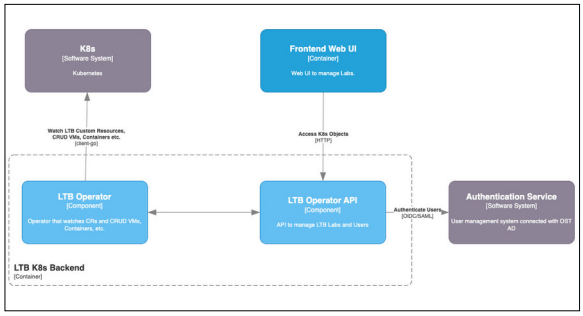
Lab Topology
Current Lab Topology Builder UI



Architecture
Own presentment



Components
Own presentment



Advisors
Urs Baumann, Yannick Zwicker

Co-Examiner
Philip Schmid, Isovalent, Wallisellen, ZH

Subject Area
Software, Networks, Security & Cloud Infrastructure