Lab Orchestrator

Marco Schlicht
RWTH Aachen

Abstract: The Lab Orchestrator is a tool that helps you to orchestrate labs. A Lab consists of multiple VMs which can be accessed by students for learning purposes. The Orchestrator allows to define such Labs by uploading VM images and instructions for the learning task. Students can then spin up the lab setup and access the VMs via NoVNC to accomplish the given task.

Motivation:

At the university, lecturers can simply provide their students with a VM in which the students can do their assignments. Software is preinstalled and preconfigured in these VMs so that students can, for example, start programming their microcontroller directly with the IDE provided. This gives the lecturers the advantage that all students use the same system and therefore only have to provide support for this system and do not have to worry about problems that differ from system to system. In addition, the VM forms a sandbox and thus changes can be made to the system in this environment and if something breaks, a snapshot is imported or the original image is reinstalled. However, the possibilities here are limited to a single VM and local deployment. It is neither possible to simply start a whole network of VMs, nor is it possible to open these VMs in the browser.

Method:

The work on the project was divided into four phases. First, the acquiring of knowledge with a proof of concept. In this phase, the basics about Docker, Kubernetes and KubeVirt were worked out which is necessary to carry out this project. Then, a proof of concept was used to show that the Lab Orchestrator idea is possible with these technologies. In the second phase a prototype was created, which reproduces the concepts of the proof of concept in a program in order to identify possible problems and solve them at an early stage. Phase 3 and 4 are respectively an alpha and beta phase of the project, in which the actual project is carried out. Here a library and an API are developed.

Discussion:

What still need to be done are some bug fixes and improvements to the api. Currently errors in the Kubernetes api are not caught there but ignored. Proper error handling can be added here. Also proper logging should be added.
We need to make a frontend that integrates the API, LabVNC and instructions. These frontend need to implement the instruction goals. Also taking a look at why the clipboard in noVNC doesn't work would be good.
Some parts are missing the documentation right now for example the LabOrchestrator-DjangoAdapter and the developer documentation in the LabOrchestratorLib. This needs to be added. The most important parts have already have unittests but not all parts. This is also something I need to add in the future to improve the software quality.

Introduction:

The Lab Orchestrator should make it possible to start a network of VMs and make them accessible via the network. In the network of VMs several VMs should run at the same time and if a user has access to one of the VMs it should be possible to access others. It should be possible to start several of such networks, so that users can work independently of each other. A user has a user session and in this session a network is started for this user, in which the user can work. The access to the VMs should be possible via an integration in the web. The user should be able to start a lab on a web page and then access the graphical user interface or the VMs in a frame or HTML Canvas to the graphical user interface or the terminal of the VMs. In addition to the integration as a frame or canvas, it should also be possible to integrate instructions. These instructions should contain several pages and several steps per page and teach the person working in the VM. Thereby the tutorial contains different features, e.g. tickable steps, to see the progress and tutorial boxes, which give knowledge and explain certain parts of the scenario. These tutorials are meant to extend the mere sandbox to teach people something.
The project should contain a library that contains the core features and makes them available in python. Another part should be an API with that the library can be used in the web.

```
GET /api/labs/
```

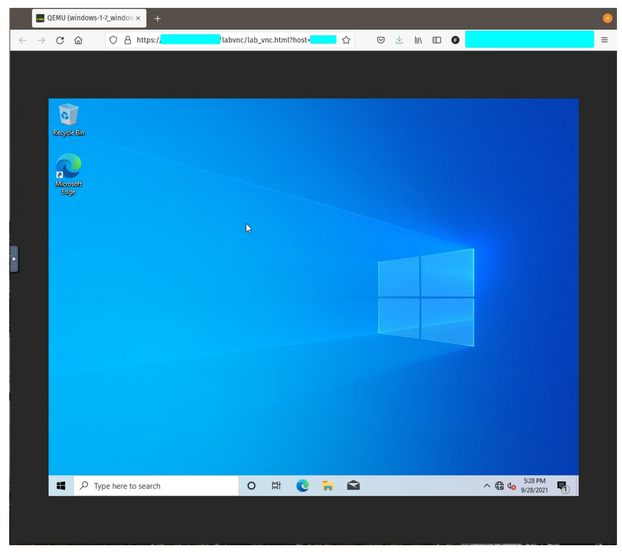
```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "lab_docker_images": [
            {
                "id": 1,
                "docker_image_name": "attacker",
                "lab": 1,
                "docker_image": 1
            },
            {
                "id": 2,
                "docker_image_name": "server",
                "lab": 1,
                "docker_image": 2
            }
        ],
        "name": "Pentesting Ubuntu",
        "namespace_prefix": "pentest-ubuntu",
        "description": "Learn how to attack a ubuntu."
    }
]
```

This Lab contains two VMs, one attacker VM and one server.

Target Groups:

- Universities
- IT Security Personal
- Companies



LabVNC connected to a windows VM.

Results:

Workplan Goal	Required	Success
Using Kubernetes with KubeVirt	No	Yes
Support Linux and Windows VM images	No	Yes
Image with VNC and terminal access from browser	Yes	Yes
Run images in a network	Yes	Yes
Multiuser support	Yes	Partly
Start and stop labs	Yes	Yes
Pause and resume labs	No	No
Add and remove labs	Yes	Yes
Configure routing	Yes	Partly
Authentification in routing	Yes	Yes
Show authentification details in labs	Yes	Yes
Connect users and their labs	Yes	Yes
Add instructions	Yes	Yes
Connect instructions and labs	Yes	Yes
Instructions: Use markdown or HTML	No	Partly
Instructions: Pages with text	Yes	Yes
Instructions: Controller to select VM	Yes	Yes
Instructions: Steps per page	No	Partly
Instructions: Tick of steps	No	No
Instructions: Progress bar	No	No
Instructions: Include pictures and other media	No	Partly
Instructions: Display knowledge texts	Yes	Partly
Instructions: Interact with VMs	No	No
Instructions: Variables	No	No
User management	Yes	Yes
API with all functionality of the library	Yes	Yes

/LabOrchestrator/LabOrchestrator

References can be found in: https://github.com/LabOrchestrator/LabOrchestrator-Documentation/blob/main/documentation.pdf