

Exercises that include multiple networked computers illustrate an advantage of using containers over VMs, namely, containers require significantly less resources than do VMs. A student laptop that struggles to run two or more VMs can readily run multiple containers simultaneously, as shown in this 50 second demonstration: <https://youtu.be/JDV6jGF3Szw>

Lab designers enhance labs to include automated assessment using directives built into the framework. For example, ten rather simple directives can evaluate the following question regarding a student's work on a lab depicted in Figure 1:

“Was there any single iptables configuration during which the student used nmap to demonstrate that:

- The remote workstation could reach the HTTPS port but not the SQL port, and,
- The local workstation could reach the HTTPS port and the SQL port.”

1.2 Obtaining the Labtainer development kit

Installation of Labtainers is described in the *Labtainer Student Guide*, which also includes instructions for installing an Ubuntu VM (if you do not already have a Linux system), and the Labtainer framework. Our website also distributes pre-packaged VM appliances that already have Labtainers installed. Labtainers will work with any Linux distribution that supports Docker containers. If you already have Docker installed on a Linux system, reference the Student Guide for other dependencies.

The difference between the development kit and the standard Labtainer distribution is primarily just the lab definition files, which are withheld from the general distribution for efficiency.

If you have a Labtainer installation (e.g., our pre-packaged VM), you can get the developer files by going to your labtainers directory, e.g., `~/labtainer/` and running `./update-designer.sh`¹ You may then want to logout and login again, or run a new `bash` shell because that script sets some environment variables.

It is suggested that you periodically run that update script to get the latest lab definition files, and to update framework software.

1.3 Content of this guide

This guide describes how to build new labs, but first, section 2 gives an overview of how students interact with Labtainers. The steps taken to create a new lab are provided in section 3, and the mechanics of defining the lab execution environment are in section 4.

Individualizing labs to discourage sharing of solutions is described in 5. Section 6 then describes how to define criteria to enable automated assessment of student work.

Networking considerations are described in 8. Section 9 covers the process of building, publishing and maintaining labs.

Strategies for creating multi-user Labtainer exercises are discussed in section 12. Section 13 identifies limitations of the framework and section 14 includes application-specific notes, e.g., notes relevant to including Firefox in a lab.

Automated testing of labs is supported using our SimLab tool as described in Appendix A.

2 Overview of the student environment and workflow

Labtainers support laboratory exercises designed for Linux environments, ranging from interaction with individual programs to labs that include what appear to be multiple components and

¹The student password for the pre-packaged VM is "password123".