

## 10 Labtainer Instructor Modules (IModules)

This guide describes how instructors can add content to Labtainers. Instructors extend Labtainers with new labs or customized versions of existing labs by defining IModules and directing their students to enable the IModules within their individual Labtainers instances.<sup>22</sup> Students simply type: `imodule <path>` to add a given URL to their Labtainers instance. The scope of instructor-generated extensions can range from modified lab manuals to new Labtainer exercises. The Labtainers framework provides tools to assist instructors in creating and publishing these extensions.

### 10.1 Labtainers distribution strategy

To understand how IModules are distributed, it is helpful to first review the general Labtainers distribution strategy. A Labtainers installation, (e.g., the initial content of a Labtainers VM appliance, or the results of installing from the distribution), includes the scripts and configuration files needed to run all Labtainers exercises. The installation initially only includes a small number of Docker container images that provide the core of container images for each of the labs. When a student first starts a given lab, the framework retrieves all Docker image layers required for that lab. These layers are retrieved from the Docker Hub, and build upon the core images present in the initial distribution. The scripts and configuration files are published as a tar archive on the Labtainers website. Whenever a Labtainers installation is updated, the archive is retrieved from the website and used to update the installation.

Files needed to create Docker images are typically not distributed in Labtainers distributions, but are installed when the user runs the update-designer script. These files are drawn from a separate tar archive on the Labtainers website.

### 10.2 Imodule distribution strategy

Instructors place archives on a web server and student instances of Labtainers retrieve those archives from the web server while retrieving other Labtainer updates. When creating new labs, instructors publish the lab Docker images to DockerHub, where they'll be retrieved by the framework when students run that lab. While the publishing of extensions does not depend on any particular source control system, supporting tools that simplify archive creation are built around git.

Archives published by instructors are tar files that include only changed and new files, relative to the Labtainers baseline. Inclusion of unchanged (relative to the Labtainers baseline) files is discouraged, as is publishing only deltas from previous IModule publications. Put another way, an IModule will contain any and all files necessary for running, (not building), all new labs – or to modify existing labs, relative to the Labtainers baseline as defined by the GitHub master repository.

Support tools simplify creation of IModule tar files through use of git attributes. Instructors who chose not to use git are responsible for creating a tar of selected files – which may be trivial, e.g., if the IModule consists of lab manual modifications or new lab guides. Paths within tar files will be relative to the labtainers/lab directory. For example, a revised telnet-lab manual would have the path:

```
telnet-lab/docs/telnet-lab.pdf
```

Note the modified source, e.g., docx files, need not be included in the IModule archive, though the support tools do include them.

---

<sup>22</sup>Or, instructors can enable IModules in VMs, and direct students to use those.