9.6 Lab versions

Substantive changes to an existing published lab should be made in a new named lab. A *substantive* change is defined as one that would break any existing installation in a manner that could not be corrected with a framework update. Issues with compatibility between two lab versions is often due to there being lab-specific files on the framework, (i.e., from where the lab is run) as well as within the Docker images that make up the lab. When a newer version of a lab image is published, it must be able to work with existing installations. If that requires an update to the framework, then that update cannot break any existing labs present in that installation, i.e., labs that have already been started.

For example, **never** change container names for existing labs. If such a change is needed, create a new lab, and assign version numbers to it and the old lab.

Lab version numbers are kept in the optional `labs/[lab]/config/version` file. There is no need to have such a file until there are two or more versions of the same lab. (Note if you want two versions of a given lab to be runnable and to appear in the list of labs, then they are not versions of the same lab. They are different labs.) The format of a lab version file is:

```
lab-base version
```

where `lab-base` is a name to associate with the multiple versions. It can be anything and does not appear at the user interface. The `version` is an integer.

To create a new version of a lab:

- Create a new lab using `new_lab_setup.py` (perhaps with the clone option).
- Create a version file for the old lab (if it does not already have one).
- Create a version file for the new lab, giving it a version numerically greater than the old version.
- Add the old lab name to the trunk/publish/skip-labs list.

When the user types the `labtainer` command with no arguments, the list will then only include the latest version of that lab. An exception is if the old lab already has been run in this installation, in which case both lab versions will display.

9.7 Creating new base images

Labtainer base images are managed using scripts and configuration files in the `scripts/designer` directory. The `bin` subdirectory includes a set of scripts that create various base images, and the `base_dockerfiles` contain their Dockerfiles. Use those as a template.

Typically, new base images are created to support a new lab. Proper Labtainer lab Dockerfiles have `FROM` directives that include the `$registry/` qualifier, however your new base image might not yet be published to a registry as you test it, and tagging the new base image with the registry name may complicate your desired workflow. Use the `-L` option to the `rebuild` command to direct the build to use unqualified image names if needed.

9.8 Importing labs: Warning!

Avoid the use of "shared folders" in VMWare and VirtualBox as a means of copying lab directories. Use tar and/or scp instead. Otherwise permissions of directories may be changed, e.g., no x access to /etc for other.