davs://my.nps.edu/webdav/c3o-staging/document_library/labtainers

After transfering the files, use the Liferay "Publish to Live" function to make the files available on the Labtainers website (which is also where they are pulled from when a student runs update-labtainer.sh).

Be sure to push your Git repository updates to the GitHub master.

The distrib/publish.py script is used to rebuild and publish individual labs, or optionally all of the Labtainer exercises managed by NPS. The publish.py (without the -l option) script will only rebuild labs that have changed. After pushing a new lab container image to the Docker hub, the script deletes the image from the local system. The intent is to ensure that future testing of the lab is done on the authoritative copy, i.e., from the hub.

Labtainer base images are built and published from the scripts/designer/bin directory. Prior to publishing baseline images, it is suggested that all local images be purged from the development machine, e,g.,

```
/trunk/setup_scripts/destroy-docker.sh
```

This will ensure that new baseline images do not incorporate layer remnants.

All new images should be first built and pushed onto the test registry, i.e., using the ./publish_image.sh <image> -t

Framework modifications made to support changed or new functions within container images must be evaluated with respect to their impact on compatibility. If a new lab image requires an updated Labtainers framework, then the "framework_version" must be incremented within the bin/labutils.py script **before** the image is built and published. This will prompt users to run update-labtainer.sh prior to running any newer lab image. Also insure that these lines are present in the container dockerfile:

```
ARG version
LABEL version=$version
```

And, be sure to publish the revised framework before publishing the revised lab(s).

9.2 Alternate registry for testing

If the environment variable TEST_REGISTRY, is set to TRUE, labs to be pulled and pushed into an alternate registry defined in the trunk/config/labtainer.config file test_registry entry. Also, the build_lab.py, labtainer, and publish.sh scripts include -t flags to force the system to reference the test registry instead of the Docker Hub. It is easy to set up a registry (it is a container!), https://docs.docker.com/registry/deploying/ Use the trunk/setup_scripts/prep-testregistry.
script to prepare a test system to use a test registry.

9.3 Large lab files

Consider storing the authoritative source of large files (e.g., pcaps) or directories in external locations, e.g., nps.box.com. This has two advantages: 1) Reduces the size of the lab designer distribution tar; 2) avoids putting large files into github. Note this issue does not affect container images, which will always include the large files regardless of how they are stored. The question is simply the location of the source of the large files for purposes of rebuilding a specific lab. Our model is to provide potential lab developers with a distribution that is not gigabytes, but also contains whatever is needed to rebuild existing labs – or at least links that are automatically followed when building the lab.

A file named <lab>/config/bigexternal.txt with entries as follows: